Sublinear Time Evaluation of Logically Defined Queries on Databases of Bounded Degree

Polly Victoria Fahey

The University of Leeds
School of Computing

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

September 2021

Declaration

The candidate confirms that the work submitted is his/her/their own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of this work in Chapters 4 and 5 have been published in the following article:

- Isolde Adler and Polly Fahey, Faster Property Testers in a Variation of the Bounded Degree Model, In 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, 2020

Some parts of this work in Chapters 4, 5 and 6 have been submitted for publication in the following articles:

- Isolde Adler and Polly Fahey, Faster Property Testers in a Variation of the Bounded Degree Model, submitted to ACM Transactions on Computational Logic
- Isolde Adler and Polly Fahey, Towards Approximate Query Enumeration with Sublinear Preprocessing Time, submitted to Journal of Logical Methods in Computer Science

All three papers are primarily the work of the second author (Polly Fahey), and the contribution of the other author (Isolde Adler) was through discussions, ideas and proof reading.

Acknowledgements

First and foremost, I would like to thank my supervisor Isolde Adler. Isolde's support and guidance has been invaluable these past four years. She has offered many interesting discussions, pointed me in the direction of useful results, and provided quality feedback on my work.

I would also like to thank the friends I have met during my PhD journey. In particular, Noleen, who proofread some of this thesis, accompanied me on many enjoyable conferences, and who was always available for a chat.

I would like to acknowledge the University of Leeds for funding my research and my examiners for their time and effort.

Last but definitely not least, I would like to thank my family. My Mum and Dad for always encouraging and supporting me. My sister Naomi for her advice, listening to my long rants, and just for being a great friend. My dog Salvador for always wanting a cuddle and walks in the woods. And finally, Christian, who has been there for me through it all. Without his consistent love and support, I would not be where I am today.

Abstract

Currently, there is an ever-growing need for extremely efficient algorithms to extract information from data. However, when the input data is huge, even reading the whole input once is too costly and hence many algorithms that are traditionally classified as 'efficient' (i.e. run in polynomial time) become impractical. This motivates the study of algorithms that run in *sublinear* time. To achieve sublinear time, it is unavoidable that we must sacrifice some accuracy (since we cannot read the whole input), but in view of many practical applications it is crucial that we provide guarantees on the accuracy of the output. In this thesis, we aim at providing sublinear time algorithms for the approximate evaluation of queries on relational databases of bounded degree, that come with probabilistic accuracy guarantees. We mainly focus on queries definable in first-order logic and monadic second-order logic with counting.

For boolean queries, we use the framework of *property testing*. In property testing, for a property **P**, the goal is to distinguish inputs that have **P** from those that are far from having **P** with high probability correctly, by querying only a small number of local parts of the input. Much research has focussed on the *query complexity* of property testing algorithms, i. e. the number of queries the algorithm makes to the input, but in view of applications, the *running time* of the algorithm is equally relevant. In the bounded degree relational database model, which was introduced in (Adler, Harwath STACS 2018) and is a natural extension of the bounded degree graph model, the distance to **P** is measured by the number of tuple modifications (additions or deletions), that are necessary to transform a database into one with property **P**. We introduce a new model, which is based on the bounded degree relational database model, but the distance measure allows both tuple modifications and element modifications. We begin an investigation of which conditions allow *constant* time property testing algorithms in our new model. In particular, we show that on databases of bounded degree and bounded tree-width, all boolean queries expressible in monadic second-order logic with counting are testable in constant time.

On the way to proving our results in the new model we also partially answer an open problem by Alon. Alon (Proposition 19.10 in: Lovász, Large networks and graph limits, 2012) proved that for every bounded degree graph $\mathscr{G}$ there exists a constant size graph $\mathscr{H}$ that has a similar neighbourhood distribution to $\mathscr{G}$. This proof is only existential and it was

suggested as an open problem to find explicit bounds on the size of $\mathscr{H}$. We find bounds on the size of $\mathscr{H}$ for two special cases.

Furthermore, we study query enumeration of non-boolean queries. In the *query enumeration problem*, after a preprocessing phase the aim is to enumerate all answers to the query with only a small delay between any two consecutive answers. We introduce a new model for the approximate query enumeration on classes of relational databases of bounded degree. We aim for *sublinear* time preprocessing and *constant* delay. Since sublinear running time does not allow reading the whole input database even once, sacrificing some accuracy is inevitable for our speed-up. Nevertheless, our enumeration algorithms come with guarantees: With high probability, (1) only tuples are enumerated that are answers to the query or 'close' to being answers to the query, and (2) if the proportion of tuples that are answers to the query is sufficiently large, then all answers will be enumerated. Here the notion of 'closeness' is a tuple edit distance in the input database. We identify conditions under which first-order definable queries can be enumerated approximately with constant delay after a sublinear preprocessing phase. In particular, we show that on databases of bounded degree and bounded tree-width, queries expressible in first-order logic can be approximately enumerated with constant delay after a sublinear preprocessing phase.

Table of contents

Chapter 1

Introduction

Given the ubiquity and sheer size of stored data nowadays, there is an immense need for highly efficient algorithms to extract information from the data. When the input data is huge, many algorithms that are traditionally classified as 'efficient' become impractical. Hence in practice often heuristics are used, at the price of losing control over the quality of the computed information. In many application areas, however, such as aviation, security, medicine, and research, accuracy guarantees regarding the computed output are crucial.

In this thesis we address this problem. We introduce new models and study existing models which allow us to design sublinear time algorithms that approximately answer queries on relational databases of bounded degree (where a database has degree at most d if each element in its domain appears in at most d tuples). The models we study enable us to decrease the running time significantly compared to traditional algorithms whilst providing probabilistic accuracy guarantees. This speed-up in running time, whilst sacrificing some accuracy, can be crucial for dealing with large inputs. In particular, it can be useful for a quick exploration of newly obtained data (e. g. biological networks). Based on the outcome of the exploration, a decision can then be taken whether to use a more time consuming exact algorithm in a second step.

We will mainly focus on approximately solving the following query problems: The *boolean query evaluation problem* and the *query enumeration problem*. In addition, we will also briefly touch on approximate versions of the *query membership testing* and *counting* problems. We will be mainly interested in relational database queries which are definable in first-order logic and monadic second-order logic with counting. There is a close link between the logics we study and SQL.

In the boolean query evaluation problem, given some boolean query q and relational database $\mathscr{D}$ the aim is to decide whether q is true in $\mathscr{D}$. To define an approximate version of this problem we use the framework of *property testing*. A *property* is simply a class

of objects. For example, each boolean relational database query q defines a property $\mathbf{P}_q$, the class of all relational databases satisfying q. Property testing algorithms (*testers*, for short) are given oracle access to the inputs, and their goal is to distinguish between inputs which have a given property $\mathbf{P}$ or are ε-far from having $\mathbf{P}$ with high probability correctly. The algorithms are parameterised by a distance measure ε, where the distance measure depends on the model. Property testers can be seen as a relaxation of the classical yes/no decision problem for $\mathbf{P}$, or if $\mathbf{P}$ is the property defined by some boolean database query q it can be seen as an approximate version of the boolean query evaluation problem for q. Testers make decisions by exploring only a small number of local parts of the input which are randomly chosen. In much of the research on property testing, the main focus is on developing testers that have *constant query complexity*, i.e. the number of queries made to the oracle does not depend on the input size. Whilst the query complexity of a tester does provide a lower bound on the running time, the actual running time can be much worse than the query complexity. In this thesis we also focus on the running times of our property testing algorithms. We use the bounded degree relational database property testing model of [4] (which is a straightforward generalisation of the bounded degree graph model [42]). We also propose a new natural model for bounded degree relational databases in which we can obtain better (constant) efficiency. For example, we show that in our new model, on relational databases of bounded degree and bounded tree-width, every property that is expressible in monadic second-order logic with counting is testable with constant query complexity and *constant* running time. Whereas in the classical model the fastest known testers for such properties run in polylogarithmic time. Our proof methods include a result by Alon [55, Proposition 19.10] that states that for every bounded degree graph $\mathscr{G}$ there exists a constant size graph $\mathscr{H}$ that has a similar neighbourhood distribution to $\mathscr{G}$ (this can easily be extended to relational databases). The proof of this result does not give explicit bounds on the size of $\mathscr{H}$, however, we obtain such bounds for some special cases.

In the query enumeration problem, we are given a database $\mathscr{D}$ and a query q, and the goal is to compute the set $q(\mathscr{D})$ of all answers to q on $\mathscr{D}$. However, the set $q(\mathscr{D})$ could be exponential in the number of free variables of q, and even bigger than $\mathscr{D}$, hence the total running time required to enumerate all answers may not be a meaningful complexity measure. Taking this into account, models for query enumeration distinguish two phases, a *preprocessing phase*, and an *enumeration phase*. Typically, in the preprocessing phase some form of data structure is computed from $\mathscr{D}$ and q, in such a way that in the enumeration phase all answers in $q(\mathscr{D})$ can be enumerated (without repetition) with only a small delay between any two consecutive answers. We focus on data complexity, i. e. we regard the query as being fixed, and the database being the input. *Efficiency* is measured both in terms of the

running time of the preprocessing phase and the *delay*, i. e. the maximum time between the output of any two consecutive answers. For the delay we can hope for constant time at best, independent of the size of the database. For the preprocessing phase, the best we can hope for regarding exact algorithms is linear time. We provide a new model for *approximately* enumerating the set of answers to queries on relational databases where we aim at *sublinear time* preprocessing and constant delay in the enumeration phase whilst providing probabilistic accuracy guarantees. Our approximate enumeration algorithms come with the following guarantees: With high probability, (1) only tuples are enumerated that are answers to the query or 'close' to being answers to the query, and (2) if the proportion of tuples that are answers to the query is sufficiently large, then all answers will be enumerated. Here the notion of 'closeness' is a tuple edit distance in the input database. We begin an investigation of under which conditions first-order definable queries can be enumerated approximately with constant delay after a sublinear preprocessing phase. In particular, we show that on databases of bounded degree and bounded tree-width, queries expressible in first-order logic can be approximately enumerated with constant delay after a sublinear preprocessing phase.

Our algorithms rely on a fixed upper bound of the degree of the input relational database. In practice relational databases/networks are often sparse, however, bounded degree is too restrictive. Hence we see our work as a first step towards efficiently querying sparse databases. Extending our work to more general classes of sparse databases (e.g classes with bounded average degree) will need new ideas as our methods rely strongly on the fixed upper bound of the degree (for example we use Hanf normal form of first-order logic). Furthermore, the constants we get in our running times are often too big for implementation, and so further research is needed to reduce these constants.

1.1 Motivation

Algorithmic meta-theorems are general algorithmic results that apply to large classes of problems on specific classes of inputs. One of the most well known algorithmic meta-theorems is Courcelle's theorem [25] that states that every property which is definable in monadic second-order logic is decidable in linear time on graphs of bounded tree-width. There has since been many other algorithmic meta-theorems (including some very recent) for example in [60, 35, 28, 31, 43, 21]. In this thesis we give results in a similar flavour but in the framework of property testing with constant running times, and for approximate query enumeration with sublinear preprocessing time (we give more details of these results in the following section).

In the research area of *approximate query processing*, the aim is to provide approximate answers to database queries at a fraction of the running time of an exact query answer. There has been much research in this area for example in [46, 38, 22, 12, 62]. We combine methods from property testing, logic and query enumeration to give results in the spirit of approximate query processing.

In property testing on graphs, as previously discussed, much of the focus is on obtaining property testing algorithms with constant query complexity. For example, it is known that in the bounded degree graph model every hyperfinite property is testable with constant query complexity [57], however, bounds on the running times cannot be obtained. In other results in property testing, for example in [17], the algorithms may have sublinear running times but it is not discussed or made clear. In this thesis we take a different approach and focus on the running times as well as the query complexity of our property testing algorithms.

1.2 Contributions

In this section, we will highlight our main contributions of this thesis. We will split this into three parts, (1) approximate boolean query evaluation, (2) finding constant size databases that preserve the local structure of large databases, and (3) approximate enumeration.

1.2.1 Approximate boolean query evaluation

Before we start let us briefly introduce the bounded degree relational database property testing model of [4], which is a generalisation of the bounded degree graph model [42]. We will call this model the BDRD model for short (for a more in depth introduction of the BDRD model we direct the reader to Section 2.6). Note that we will sometimes refer to relational databases as just databases.

Formally, a *property* **P** is an isomorphism closed class of relational databases. The BDRD model assumes a uniform upper bound d on the degree of the input databases. For $\varepsilon \in [0,1]$, a database $\mathscr{D}$ with domain of size n is ε-*close* to satisfying **P**, if we can make $\mathscr{D}$ isomorphic to a member of **P** by editing (inserting or removing) at most εdn tuples in relations of $\mathscr{D}$ (i. e. at most an 'ε-fraction' of the maximum possible number dn of tuples in relations). Otherwise, $\mathscr{D}$ is called ε-*far* from **P**. Property testing algorithms do not have access to the whole input $\mathscr{D}$, but instead are given access via an oracle. In the BDRD model testing algorithms can make *oracle queries* of the form 'what is the j-th tuple of the relation $R^{\mathscr{D}}$ which contains the i-th element of the domain?'. Note that we assume a linear order on the elements of the domain of a database and we assume that oracle queries can be answered in constant time.

Let $\mathbf{C}$ be some class of d-degree bounded databases and let $\mathbf{P} \subseteq \mathbf{C}$ be a property on $\mathbf{C}$. An *ε-tester* for $\mathbf{P}$ on $\mathbf{C}$ is then a probabilistic algorithm which receives the size n of the domain of the input database $\mathscr{D} \in \mathbf{C}$, and has oracle access to $\mathscr{D}$. The ε-tester does the following:

1. If $\mathscr{D}$ has property $\mathbf{P}$ (i. e. $\mathscr{D}$ is a member of $\mathbf{P}$), then the tester accepts with probability at least $2/3$.

2. If $\mathscr{D}$ is ε-far from $\mathbf{P}$, then the tester rejects with probability at least $2/3$.

We say that $\mathbf{P}$ is *uniformly testable* on $\mathbf{C}$, if for every $\varepsilon \in (0,1]$ there exists an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ that has constant query complexity i.e. the number of queries made to the oracle does not depend on the input size.

Testing properties definable in first-order logic The main content of this thesis starts in Chapter 3, where we consider the testability of properties definable in two different fragments of first-order logic (FO). We first note that any property that is definable by a sentence in the existential fragment of FO (i.e. sentences of the form $\exists x_1, \ldots \exists x_\ell \psi(x_1, \ldots, x_\ell)$ where ψ is quantifier free) is trivially uniformly testable in the BDRD model. We then show the following.

Every property that is definable by a sentence in the universal fragment of FO (i.e. sentences of the form $\forall x_1, \ldots \forall x_\ell \psi(x_1, \ldots, x_\ell)$ where ψ is quantifier free) is uniformly testable in the BDRD *model in constant time (Theorem 3.3).*

To prove Theorem 3.3 we first note that testing properties definable by a sentence in the universal fragment of FO is equivalent to testing for the absence of induced sub-databases isomorphic to some database from a finite set. Goldreich and Ron [42] proved that on bounded degree graphs testing for the absence of (not necessarily induced) subgraphs isomorphic to a graph from a finite set is uniformly testable in constant time. They start by proving that the property of subgraph-freeness is uniformly testable in constant time and then prove that the union of (finitely many) such properties are also uniformly testable. We modify their tester for testing subgraph-freeness to test for induced subgraph-freeness and translate this to databases. We then use different techniques from [42] to prove that the union of (finitely many) induced sub-database-freeness properties are uniformly testable in the BDRD model in constant time.

A new property testing model In Chapter 5 we propose a new model for property testing on bounded degree relational databases, which we call the $\text{BDRD}_{+/-}$ model. In the $\text{BDRD}_{+/-}$ model the distance measure allows both tuple deletions and insertions, and *deletion and insertion of elements of the domain*. On graphs, this translates to edge insertions and

deletions, and *vertex insertions and deletions*. We argue that this yields a natural distance measure. Indeed, take any (sufficiently large) graph $\mathscr{G}$, and let $\mathscr{H}$ be obtained from $\mathscr{G}$ by adding an isolated vertex. Then $\mathscr{G}$ and $\mathscr{H}$ are ε-far for every $\varepsilon \in (0,1]$ under the classical distance measure, although they only differ in one vertex. In contrast, our distance measure allows for a small number of vertex modifications. While comparing graphs on different numbers of vertices by adding isolated vertices was done implicitly as part of the study the testability of outerplanar graphs [13], to the best of our knowledge, such a distance measure has not been considered before as part of a model in property testing, which seems surprising to us.

Formally, in the $\text{BDRD}_{+/-}$ model, two databases $\mathscr{D}$ and $\mathscr{D}'$ are ε-*close*, if they can be made isomorphic by at most εdn *modifications*, where a modification is either, (1) removing a tuple from a relation, (2) inserting a tuple to a relation, (3) removing an element from the domain (and, as a consequence, any tuple containing that element is removed), or (4) inserting an element into the domain. Here n is the minimum of the sizes of the domains of $\mathscr{D}$ and $\mathscr{D}'$. In Section 5.1 we give the full details of our model. We note that the $\text{BDRD}_{+/-}$ model differs from the BDRD model only in the choice of the distance measure. While we work in the setting of relational databases, we would like to emphasize that our results carry over to (undirected and directed) graphs, as these can be seen as special instances of relational databases.

We show that the $\text{BDRD}_{+/-}$ model is in fact stronger than the BDRD model: Any property testable in the BDRD model is also testable in the $\text{BDRD}_{+/-}$ model with the same query complexity and running time (Lemma 5.3), but there are examples that show that the converse is not true (Lemma 5.5).

It is known that in the bounded degree graph model, every hyperfinite graph property is (non-uniformly in n) testable [57] with constant query complexity. However, no bound on the running time can be obtained in this general setting. Indeed, there exist hyperfinite properties (of edgeless graphs) that are uncomputable. In [4], Adler and Harwath ask which conditions guarantee both low query complexity *and* efficient running time. They prove a meta-theorem stating that, on classes of databases (or graphs) of bounded degree and bounded tree-width, every property that can be expressed by a sentence of monadic second-order logic with counting (CMSO) is uniformly testable with *constant* query complexity and *polylogarithmic* running time in the BDRD model. Treating many algorithmic problems simultaneously, this can be seen as an algorithmic *meta-theorem* within the line of research inspired by Courcelle's famous theorem [25]. CMSO extends FO and hence properties expressible in FO (e.g. subgraph/sub-database-freeness) are also expressible in CMSO. Other examples of graph properties expressible in CMSO include bipartiteness, colourability, even-hole-

freeness and Hamiltonicity. Rigidity (i. e. the absence of a non-trivial automorphism) cannot be expressed in CMSO (cf. [26] for more details).

We show the following.

In the $\mathrm{BDRD}_{+/-}$ *model, on classes of databases (or graphs) of bounded degree and bounded tree-width, every property that can be expressed by a sentence of CMSO is uniformly testable with* constant *query complexity and* constant *running time (Theorem 5.15).*

The question of whether constant running time can also be achieved in the BDRD model remains open. For proving Theorem 5.15, we give a general condition under which properties are uniformly testable in constant time in the $\mathrm{BDRD}_{+/-}$ model. To describe this condition let us first briefly introduce some terminology. A property **P** is *hyperfinite* on a class of databases **C** if every database in **P** can be partitioned into connected components of constant size by removing only a constant fraction of the tuples such that the resulting partitioned database is in **C**. For $r \in \mathbb{N}$ and an element a in the domain of a database $\mathscr{D}$, the *r-neighbourhood type* of a in $\mathscr{D}$ is the isomorphism type of the sub-database of $\mathscr{D}$ induced by all elements that are at distance at most r from a in the underlying graph of $\mathscr{D}$, expanded by a. The *r-histogram* of a bounded degree database $\mathscr{D}$, denoted by $\mathrm{h}_r(\mathscr{D})$, is a vector indexed by the r-neighbourhood types, where the component corresponding to the r-neighbourhood type τ contains the number of elements in $\mathscr{D}$ that realise τ. The *r-neighbourhood distribution* of $\mathscr{D}$ is the vector $\mathrm{h}_r(\mathscr{D})/n$ where $\mathscr{D}$ is on n elements. We show that for any property **P** and input class **C** (which is closed under removing tuples, removing elements and inserting elements), if **P** is hyperfinite on **C** and the set of r-histograms of the databases in **P** are semilinear, then **P** is uniformly testable on **C** in constant time (Theorem 5.14). From a result in [34] about many-sorted spectra of CMSO sentences it can be derived that the set of r-histogram vectors of properties defined by a CMSO sentence on the class of all bounded degree and bounded tree-width databases are semilinear [4]. It is also known that the class of all bounded degree and bounded tree-width databases are hyperfinite and hence any property is hyperfinite on the class of all bounded degree and bounded tree-width databases. Furthermore, it is easy to show that the class of all bounded degree and bounded tree-width databases is closed under removing tuples, removing elements and inserting (isolated) elements. Hence as a corollary we then obtain Theorem 5.15 that every property definable by a CMSO sentence is uniformly testable on the class of databases with bounded degree and bounded tree-width in constant time.

We also discuss the constant time testability of hyperfinite hereditary properties and hyperfinite monotone properties in the BDRD and $\mathrm{BDRD}_{+/-}$ models. To the best of our knowledge, it has not been shown explicitly that hyperfinite hereditary properties are uniformly testable in constant time (in the bounded degree graph or BDRD models). In [17] it

is proved that every monotone hyperfinite property is constant time testable in the bounded degree graph model. In [27] the authors prove that hereditary properties are testable in constant time on classes of non-expanding hereditary properties (which include hyperfinite hereditary classes) in the bounded degree graph model. We sketch a proof that hyperfinite hereditary properties are uniformly testable in constant time in the BDRD model (and hence in the $\text{BDRD}_{+/-}$ model) using methods similar to [17] and [27]. We then give alternative proofs showing that hyperfinite hereditary properties are uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model and that hyperfinite monotone properties are uniformly testable in constant time in the BDRD model (and hence in the $\text{BDRD}_{+/-}$ model) using different techniques, similar to those used for Theorem 5.15.

In the future, it would be interesting to obtain a characterisation of the properties that are (efficiently) testable in the $\text{BDRD}_{+/-}$ model.

1.2.2 Constant size databases that preserve the local structure of large databases

Alon [55, Proposition 19.10] proved that for every bounded degree graph $\mathscr{G}$ there exists a constant size graph $\mathscr{H}$ that has a similar neighbourhood distribution to $\mathscr{G}$ (and this result easily extends to relational databases). However, the proof is based on a compactness argument and does not give an explicit upper bound on the size of $\mathscr{H}$. Finding such a bound was suggested by Alon as an open problem [47]. Currently, the only known bounds are for graphs with high-girth [32]. In Chapter 4 we obtain explicit bounds on the size of $\mathscr{H}$ for 'semilinear' properties, i. e. properties, where the histogram vectors of the neighbourhood distributions form a semilinear set (Theorem 4.2), and for hyperfinite databases (Theorem 4.6). Furthermore, in the semilinear case we show that $\mathscr{H}$ also belongs to the semilinear property. We believe these results are of independent interest, but Theorem 4.2 is also essential in the proof of Theorem 5.14 which states that for any property **P** and input class **C** (which is closed under removing tuples, removing elements and inserting elements), if **P** is hyperfinite on **C** and the set of r-histograms of the databases in **P** are semilinear, then **P** is uniformly testable on **C** in constant time.

1.2.3 Approximate enumeration

In Chapter 6 we consider databases $\mathscr{D}$ of bounded degree d, and we identify conditions under which FO definable queries can be enumerated approximately with constant delay after a sublinear preprocessing phase. We consider two different categories of FO definable queries, *local* and *general* (including *non-local*) queries. A FO query is *local* if, given any

bounded degree database and tuple, it can be decided by only looking at the local (fixed radius) neighbourhood around the tuple whether the tuple is an answer to the query for the database. We show the following.

On input databases of bounded degree, every (fixed) local FO definable query can be enumerated approximately with constant delay after a constant time preprocessing phase (Theorem 6.13).

On input databases of bounded degree and bounded tree-width, every (fixed) FO definable query can be enumerated approximately with constant delay after a sublinear time preprocessing phase (Theorem 6.19).

We give generalisations of the two theorems above (Theorems 6.14, 6.20 and 6.28), which we will discuss below. We also give applications of our approach to the query membership testing and counting computational problems on databases (Theorems 6.30 and 6.31).

First, let us give some more details. For any local FO query q, bounded degree database $\mathscr{D}$ and tuple $\bar{a}$ from $\mathscr{D}$ it can be decided in constant time whether $\bar{a}$ is an answer to q on $\mathscr{D}$ (Lemma 6.7). Using this fact, we show that for any fixed local FO definable query $q(\bar{x})$ with $|\bar{x}| =: k$ and $\gamma \in (0,1)$, there exists an enumeration algorithm with constant preprocessing time and constant delay, that is given a bounded degree database $\mathscr{D}$ with domain of size n as input and does the following. It enumerates a set of tuples that are answers to q on $\mathscr{D}$, and with high probability it enumerates *all* answers to q on $\mathscr{D}$ if the size of the answer set of q on $\mathscr{D}$ is larger than γn^k (i.e. the number of answers to the query is larger than a fixed fraction of the total possible number of answers).

Towards reducing the minimum size of the answer set required to enumerate *all* answers to the query, we show we actually only require size γn^c, where c is the maximum number of connected components in the neighbourhood (of some fixed radius) of an answer to q (Theorem 6.14). We argue that in practice, c can be expected to be low for natural queries.

If a FO query q is non-local, then for a database $\mathscr{D}$ and a tuple $\bar{a}$, we can no longer decide in constant time whether $\bar{a}$ is an answer to q on $\mathscr{D}$. However, using Hanf-locality of FO [44] and a result from the area of property testing, we can approximately enumerate any FO definable query on bounded degree and bounded tree-width databases with polylogarithmic preprocessing time and constant delay (Theorem 6.19). Let us now explain our notion of approximation, which is based on neighbourhood types.

For $d \in \mathbb{N}$, let $\mathbf{C}$ be a class of databases of degree at most d over a fixed finite schema. In Subsection 1.2.1 we defined the r-neighbourhood type of an element of the domain of a database $\mathscr{D} \in \mathbf{C}$. This can be extended to define the r-neighbourhood type of a *tuple* $\bar{a}$ in $\mathscr{D}$, by considering the isomorphism type of the sub-database induced by the union of the r-neighbourhoods of all components of $\bar{a}$, expanded by $\bar{a}$. We call such an isomorphism

type an *r-type (with $|\bar{a}|$ centres)*. Given a database query $q(\bar{x})$ with $|\bar{x}| =: k$ and a database $\mathscr{D}$ with domain of size n we say that a tuple $\bar{a}$ from $\mathscr{D}$ is *ε-close to being an answer of q on $\mathscr{D}$ and* $\mathbf{C}$, if $\mathscr{D}$ can be modified with tuple modifications (insertions and deletions) into a database $\mathscr{D}' \in \mathbf{C}$ with at most εdn modifications, such that $\bar{a}$ is an answer of q on $\mathscr{D}'$ and $\bar{a}$ has the same r-neighbourhood type (for some r) in $\mathscr{D}'$ and $\mathscr{D}$. We let $q(\mathscr{D}, \mathbf{C}, \varepsilon)$ be the set of k-tuples $\bar{a}$ of elements of $\mathscr{D}$ that are ε-close to being an answer of q on $\mathscr{D}$ and $\mathbf{C}$. Note that for any local first-order query q, $q(\mathscr{D}, \mathbf{C}, \varepsilon) = q(\mathscr{D})$.

We say that the enumeration problem $\mathrm{Enum}_{\mathbf{C}}(q)$ for q on $\mathbf{C}$ can be solved *approximately* with preprocessing time $H(n)$ and constant delay for answer threshold function $f(n)$, if for every $\varepsilon \in (0,1]$, there exists an algorithm, which is given oracle access to an input database $\mathscr{D} \in \mathbf{C}$ (for each given element of the domain, the tester can query the oracle for tuples in any of the relations containing the element, and we assume that oracle queries are answered in constant time), and is given the number n of elements of the domain, and proceeds in two phases. First, a preprocessing phase that runs in time $H(n)$, followed by an enumeration phase where a set S of pairwise distinct tuples is enumerated, with constant delay between any two consecutive tuples. In addition, we require that with probability at least $2/3$, (1) $S \subseteq q(\mathscr{D}) \cup q(\mathscr{D}, \mathbf{C}, \varepsilon)$, and (2) if $|q(\mathscr{D})| \geq f(n)$, then $q(\mathscr{D}) \subseteq S$.

We consider database queries that are expressible in FO. Note that our notion of approximation is designed specifically for FO queries and sparse databases and for other classes of queries and input databases alternative models may be necessary. We prove that for every FO query $q(\bar{x})$ with $|\bar{x}| = k$ the problem $\mathrm{Enum}_{\mathbf{C}_d^t}(q)$ (where $\mathbf{C}_d^t$ is the class of all databases of d-bounded degree and t-bounded tree-width) can be solved approximately with polylogarithmic preprocessing time and constant delay with answer threshold function $f(n) = \gamma n^k$ for any $\gamma \in (0,1)$ (Theorem 6.19). As with local queries, we further prove that we can actually reduce the answer threshold function to $f(n) = \gamma n^c$ where $c \leq k$ is the maximum number of connected components in the neighbourhood (of some fixed radius) of an answer to q (Theorem 6.20). We also identify a condition that is based on Hanf-locality of FO [44], which we call *Hanf-sentence testability*, and we prove a general theorem (Theorem 6.28), that for every FO query $q(\bar{x})$ with $|\bar{x}| = k$ that is Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$, the problem $\mathrm{Enum}_{\mathbf{C}}(q)$ can be solved approximately with preprocessing time $\mathscr{O}(H(n))$ and constant delay for answer threshold function $f(n) = \gamma n^c$ as above.

We illustrate our model throughout Chapter 6 with a running example which can be motivated by the problems of subgraph matching and inexact subgraph matching in social and biological networks (e.g. [66, 63]). We show that our running example is Hanf-sentence testable on the class of all bounded degree graphs in constant time, and hence by Theorem 6.28 it can be approximately enumerated with constant preprocessing time.

We further show that if we use the distance measure used in the $\mathrm{BDRD}_{+/-}$ model (rather than the distance measure used in the BDRD model) we can obtain approximate enumeration algorithms for general first-order queries that have constant (rather than polylogarithmic) preprocessing time and constant delay (Theorem 6.35).

1.3 Related work

In this section we will discuss related work on property testing and query enumeration.

1.3.1 Property testing

The notion of property testing was first introduced in 1996 by Rubinfeld and Sudan [58], motivated by program checking. In their notion of property testing their property testing algorithms distinguish with high probability correctly between functions that have some predefined property from those that are far away from having the property. The distance between two functions is measured by the fraction of elements in the domain on which the functions disagree. They note that the existence of a robust characterisation implies the existence of a property tester. The main aim of their work is then to find robust characterisations for the family of low degree univariate and multivariate polynomials.

Dense graph model

Goldreich, Goldwasser and Ron in 1996, in an extended abstract of [40], then extended the notion of property testing to testing graph properties. In their model a graph $\mathscr{G}$ is represented by the function $g : V \times V \to \{0,1\}$ where V are the vertices of $\mathscr{G}$ and $g(u,v) = 1$ if and only if there is an edge between u and v in $\mathscr{G}$, i.e. $\mathscr{G}$ is represented by its adjacency matrix. The testing algorithms are given oracle access to an input graph $\mathscr{G}$, i.e. to the function g, and hence the tester can make queries to the oracle of the form 'is there an edge between vertices u and v?'. In this model two n-vertex graphs $\mathscr{G}$ and $\mathscr{H}$ are ε-far if you need to insert or remove more than εn^2 edges to $\mathscr{G}$ and $\mathscr{H}$ to make them isomorphic. A graph is then ε-far from a graph property (an isomorphism closed class of graphs) if it is ε-far from every graph in the property. This model is most suitable for dense graphs and hence it is called the *dense graph model*.

Goldreich, Goldwasser and Ron in [40] proved that in the dense graph model all general partition graph properties (such as k-colourability, cliques etc.) are testable. Alon, Fischer, Krivelevich and Szegedy [6] began a logical characterisation of graph properties that are testable in the dense graph model. They showed that all graph properties that can be defined

by a $\exists\forall$ FO sentence are testable and some $\forall\exists$ FO sentences are not testable. There was a trend of using Szemerédi's Regularity Lemma (e.g. in [6]) and this was shown to be no coincidence by Alon, Fischer, Newman and Shapira in [7]. Szemerédi's Regularity Lemma states that for any large enough graph its vertices can be partitioned into a bounded number of parts such that the edges between most of the different parts behave randomly. Alon et al. [7] made a combinatorial characterisation, with their main result, that a graph property is testable if and only if it is regular reducible (i.e. if testing the property can be reduced to the property of testing for one of the finitely many Szemerédi-partitions). Hence the dense graph model is actually fairly well understood.

Bounded degree graph model

Given a graph $\mathscr{G}$ on n vertices with degree at most $d \in \mathbb{N}$, if n is large enough, then in the dense graph model $\mathscr{G}$ is ε-close to every other graph on n vertices with degree at most d. Therefore an alternative model is needed that is suitable for bounded degree graphs. Goldreich and Ron in 1997 [41] introduced the bounded degree graph model. In the bounded degree model an n-vertex graph $\mathscr{G}$ with degree at most d is represented by the function $g : [n] \times [d] \to \{0, 1, \ldots, n\}$, where we assume the vertex set of $\mathscr{G}$ is $[n]$ and $g(v,i) = u \in [n]$ if u is the i-th neighbour of v in $\mathscr{G}$ and $g(v,i) = 0$ if v has less than i neighbours, i.e $\mathscr{G}$ is represented by its adjacency list. Testing algorithms are then given oracle access to an input graph, i.e. to the function g, and hence the testing algorithms can make queries to the oracle of the form 'what is the i-th neighbour of vertex v?'. In this model, two n-vertex graphs $\mathscr{G}$ and $\mathscr{H}$ are ε-far if you need to insert or remove more than εnd edges to $\mathscr{G}$ and $\mathscr{H}$ to make them isomorphic. A graph is then ε-far from a graph property if it is ε-far from every graph in the property.

Goldreich and Ron in [41] showed that the following graph properties are testable in the bounded degree model; connectivity, k-edge-connectivity, k-vertex-connectivity, Eulerianity, planarity and cycle-freeness. They also developed $\Omega(\sqrt{n})$ lower bounds on the query complexity of bipartiteness and expander properties.

Other than results on testing for specific graph properties there are some testability results where either the graph properties are restricted or the inputs come from some restricted class of graphs. For example every minor-closed property is testable [17, 45], every hereditary property on hereditary non-expanding graphs is testable [27] and every hereditary property is (non-uniformly in n) testable [57].

Obtaining a characterisation of constant query testable properties is a long-standing open problem. Ito et al. [48] give a characterisation of the 1-sided error constant query testable monotone and hereditary graph properties in the bounded degree (directed and undirected)

graph model. Fichtenberger et al. [33] give a result on the combinatorial structure of every testable property, they show that every constant query testable property in the bounded degree graph model is either finite or contains an infinite hyperfinite subproperty.

In general, the bounded degree graph model is much less understood than the dense graph model.

Property testing on relational databases

As already mentioned, Adler and Harwath [4] introduced the BDRD model and showed that in this model, on classes of relational databases with bounded degree and bounded tree-width, any property that is definable in monadic second-order logic with counting is testable in polylogarithmic time. Other than the work in [4] there are only a handful of results on relational databases that utilise models from property testing. Chen and Yoshida [24] study a model which is close to the general graph model (cf. e. g. [8]) in which they study the testability of homomorphism inadmissibility. Ben-Moshe et al. [16] study the testability of near-sortedness (a property of relations that states that most tuples are close to their place in some desired order). A *conjunctive query* (CQ) is a FO formula constructed from atomic formulas using conjunctions and existential quantification only. CQ evaluation is closely related to solving constraint satisfaction problems (CSPs) [53]. CSPs have been studied under different models from property testing ([23, 65, 5]).

1.3.2 Query enumeration

Recent research has been very successful in providing exact enumeration algorithms for logically defined queries on *sparse* relational databases. In 2007, Durand and Grandjean showed that on relational databases of bounded degree, every FO query can be enumerated with constant delay after a linear time preprocessing phase [29]. This result triggered a number of papers on the exact enumeration of FO queries on relational databases with bounded degree [50], low degree [30], locally bounded expansion [61] and bounded expansion [51]. These works culminated in Schweikardt, Segoufin and Vigny's result that on *nowhere dense* databases, FO queries can be enumerated with constant delay after a pseudo-linear time preprocessing phase [59]. Monadic second-order logic queries can be enumerated with linear preprocessing time and constant delay on databases of bounded tree-width [14, 52].

There are also numerous results in the dynamic setting (where databases may be updated by inserting or removing tuples). Berkholz, Keppeler and Schweikardt [18] show that for bounded degree relational databases and FO queries, a data structure can be constructed in linear time that can be updated in constant time when a tuple is inserted or deleted from the

database. After each update to the database, the data structure allows all answers to the query to be enumerated with constant delay. A similar result was obtained by Vigny [64] for FO queries and databases of low degree.

1.4 Organisation

The rest of this thesis is organised as follows.

1. In Chapter 2 we introduce the relevant notions used throughout this thesis.
2. In Chapter 3 we consider the testability of properties definable in the existential and universal fragments of first-order logic.
3. In Chapter 4 we find explicit bounds on the size of the small graphs/databases from Alon's theorem for graphs/databases from properties whose neighbourhood histograms form a semilinear set and for hyperfinite graphs/databases.
4. In Chapter 5 we introduce our new property testing model, the $\mathrm{BDRD}_{+/-}$ model. We prove that properties definable in monadic second-order logic with counting on databases of bounded degree and bounded tree-width are uniformly testable in constant time in the $\mathrm{BDRD}_{+/-}$ model. We also discuss the constant time testability of hyperfinite monotone properties and hyperfinite hereditary properties in both the BDRD and $\mathrm{BDRD}_{+/-}$ models.
5. In Chapter 6 we introduce a new model for the approximate enumeration of first-order definable queries on databases of bounded degree. We show that on databases of bounded degree, (1) every local first-order definable query can be enumerated approximately with constant preprocessing time and constant delay, and (2) every (general) first-order definable query can be enumerated approximately with polylogarithmic preprocessing time and constant delay.
6. In Chapter 7 we give a conclusion and a discussion on future work.

Some of the work in Chapters 4 and 5 appear in [3, 1] ([1] is the journal version of [3]). Some of the work in Chapter 6 appear in [2].

Chapter 2

Preliminaries

In this chapter we will define the key notions used throughout this thesis. Note that there are some notions that are only used in certain chapters and these will be given in the respective chapter.

We start the chapter in Section 2.1 with some general set notation. In Section 2.2 we discuss our model of computation. In Sections 2.3 and 2.4 we introduce the relevant notions on graphs and relational databases respectively. In Section 2.5 we introduce first-order logic and monadic second-order logic with counting, which are the main database query languages used in this thesis. In Section 2.5 we also introduce Hanf normal form of first-order logic. In Section 2.6 we introduce the bounded degree relational database property testing model. Finally, in Section 2.7 we define semilinear sets and give some sets which are known to be semilinear which will be used in future chapters.

2.1 Set notation

We let $\mathbb{N}$ be the set of natural numbers including 0, and we denote $\mathbb{N} \setminus \{0\}$ by $\mathbb{N}_{\geq 1}$. For each $n \in \mathbb{N}_{\geq 1}$, we let $[n] = \{1, 2, \ldots, n\}$.

2.2 Model of computation

We use Random Access Machines (RAMs) and a uniform cost measure when analysing our algorithms, i. e. we assume all basic arithmetic operations including random sampling can be done in constant time, regardless of the size of the numbers involved. We assume that if we initialise an array, all entries are set to 0 and this can be done in constant time for any length or dimension array. This is achieved by using the lazy array initialisation technique

(cf. e.g. [56]) where entries are only actually stored when they are first needed. Let $k \in \mathbb{N}_{\geq 1}$ and $\mathbf{A}$ be a k-dimensional array. We assume that for a tuple $(a_1, a_2, \ldots, a_k) \in \mathbb{N}_{\geq 1}^k$, the entry $\mathbf{A}[a_1, a_2, \ldots, a_k]$ at position $(a_1, \ldots, a_k)$ can be accessed in constant time.

2.3 Graphs

In this section we will give a short introduction to the relevant graph theory notions used throughout this thesis. In this thesis we will assume that all graphs are simple graphs (unless otherwise stated).

Definition 2.1 (Simple graph)**.** *A* simple graph $\mathscr{G}$ *is a tuple* $\mathscr{G} = (V, E)$ *where* V *is a set of* vertices *and* E *is a set of* 2*-element subsets of* V *(the* edges *of* $\mathscr{G}$*). We will often denote* V *by* $V(\mathscr{G})$ *and* E *by* $E(\mathscr{G})$*. Furthermore, for an edge* $\{u, v\} \in E$ *we simply write* uv*.*

Let $\mathscr{G}$ be a graph. We say a vertex $v \in V(\mathscr{G})$ is *incident* to an edge $e \in E(\mathscr{G})$, if $v \in e$. Two vertices $u, v \in V(\mathscr{G})$ are *neighbours* if both u and v are incident to the same edge. Let $uv \in E(\mathscr{G})$ then we let $\mathscr{G} \setminus uv$ denote the graph obtained from $\mathscr{G}$ by removing the edge uv from $E(\mathscr{G})$. A *path* between the two vertices $u, v \in V(\mathscr{G})$ of length m is a sequence of distinct vertices $v_0, v_2, \ldots, v_m$ from $V(\mathscr{G})$ such that $u = v_0$, $v = v_m$ and $v_i v_{i+1} \in E(\mathscr{G})$ for all $i \in \{0, \ldots, m-1\}$. Two graphs $\mathscr{G}$ and $\mathscr{H}$ are *isomorphic* if there is a bijective map $f : V(\mathscr{G}) \to V(\mathscr{H})$ such that for every $u, v \in V(\mathscr{G})$, $uv \in E(\mathscr{G})$ if and only if $f(u)f(v) \in V(\mathscr{H})$. A graph $\mathscr{H}$ is a *subgraph* of a graph $\mathscr{G}$ if $V(\mathscr{H}) \subseteq V(\mathscr{G})$ and $E(\mathscr{H}) \subseteq E(\mathscr{G})$. A graph $\mathscr{H}$ is an *induced subgraph* of a graph $\mathscr{G}$ if $V(\mathscr{H}) \subseteq V(\mathscr{G})$ and $E(\mathscr{H}) = \{uv \in E(\mathscr{G}) \mid u, v \in V(\mathscr{H})\}$. A graph $\mathscr{G}$ is *connected* if there is a path between every pair of vertices in $\mathscr{G}$. The *connected components* of a graph $\mathscr{G}$ are the maximal connected induced subgraphs of $\mathscr{G}$.

We will now define five different classes of graphs that will be used in the following sections and chapters.

Definition 2.2 (Classes of graphs with bounded degree)**.** *Let* $\mathscr{G} = (V, E)$ *be a graph. The* degree of a vertex $v \in V$*, denoted* $\deg_{\mathscr{G}}(v)$*, is the number of edges in* E *incident to* v*. The* degree of the graph $\mathscr{G}$*, denoted* $\deg(\mathscr{G})$*, is the maximum degree of all vertices in* V*.*

A class of graphs $\mathbf{C}$ *has* bounded degree $d \in \mathbb{N}$ *if for all* $\mathscr{G} \in \mathbf{C}$*,* $\deg(\mathscr{G}) \leq d$*.*

We next define classes of graphs with *bounded tree-width*, but first we need to recall the notion of *tree decomposition* and the *width* of a tree decomposition. A graph is a *tree* if there is exactly one path between every pair of vertices in the graph. A *tree decomposition* of a graph $\mathscr{G} = (V, E)$ is a pair (X, T), where $X = \{X_1, \ldots, X_k\}$ is a family of non-empty subsets of V (often called *bags*) and T is a tree whose set of vertices is exactly X such that:

- $\bigcup_{1\leq i\leq k} X_i = V$,
- for every edge $uv \in E$ there exists some $i \in [k]$ such that $u, v \in X_i$, and
- for every $v \in V$, if $v \in X_i \cap X_j$ for some $i, j \in [k]$, then every vertex X_s on the unique path from X_i to X_j in T also contains v.

The *width* of the tree decomposition (X,T) is $\max\{|X_1|,\ldots,|X_k|\}-1$.

Definition 2.3 (Classes of graphs with bounded tree-width)**.** *Let $\mathscr{G} = (V,E)$ be a graph. The* tree-width *of $\mathscr{G}$ is the minimal width of all its tree decompositions.*

A class of graphs $\mathbf{C}$ *has* bounded tree-width $t \in \mathbb{N}$ *if for all $\mathscr{G} \in \mathbf{C}$, the tree-width of $\mathscr{G}$ is at most t.*

Definition 2.4 (Hereditary graph classes)**.** *A class of graphs is* hereditary *if it is closed under the removal of vertices (i.e. it is closed under induced subgraphs).*

Let $\mathscr{G}$ and $\mathscr{H}$ be graphs. We say $\mathscr{G}$ is *induced $\mathscr{H}$-free* if it contains no induced subgraph isomorphic to $\mathscr{H}$. For a family of graphs F, we say $\mathscr{G}$ is *induced F-free* if it contains no induced subgraph isomorphic to any graph in F. All hereditary graph classes have a (possibly infinite) set of forbidden induced subgraphs (e.g. see [10]). For a hereditary graph class $\mathbf{C}$, the *set of forbidden induced subgraphs F of* $\mathbf{C}$ is the minimal set of graphs such that for any graph $\mathscr{G}$, $\mathscr{G} \in \mathbf{C}$ if and only if $\mathscr{G}$ is induced F-free. In other words, a graph $\mathscr{H}$ belongs to F if $\mathscr{H} \notin \mathbf{C}$, but any induced subgraph of $\mathscr{H}$ is in $\mathbf{C}$.

Definition 2.5 (Monotone graph classes)**.** *A class of graphs is* monotone *if it is closed under the removal of vertices and edges (i.e. it is closed under subgraphs).*

Definition 2.6 (Hyperfinite graph classes)**.** *Let $\varepsilon \in [0,1]$ and let $k \in \mathbb{N}$. A graph $\mathscr{G} = (V,E)$ is* (ε,k)-hyperfinite *if by removing at most $\varepsilon|V|$ edges from $\mathscr{G}$ you can obtain a graph whose connected components have size at most k. For a function $\rho : [0,1] \to \mathbb{R}^+$, a graph $\mathscr{G}$ is* ρ-hyperfinite *if $\mathscr{G}$ is $(\varepsilon,\rho(\varepsilon))$-hyperfinite for every $\varepsilon \in [0,1]$.*

A class of graphs $\mathbf{C}$ *is* hyperfinite *if there exists a function ρ such that every graph in $\mathbf{C}$ is ρ-hyperfinite.*

2.4 Relational databases

In this thesis we are interested in evaluating queries on *relational databases*. In this section we will define all the relevant notions for relational databases.

Definition 2.7 (Relational database)**.** *A* schema *is a finite set $\sigma = \{R_1, \dots, R_{|\sigma|}\}$ of relation names, where each $R \in \sigma$ has an* arity *$ar(R) \in \mathbb{N}_{\geq 1}$. A* relational database *$\mathscr{D}$ of schema σ (σ-db for short) is of the form $\mathscr{D} = (D, R_1^{\mathscr{D}}, \dots, R_{|\sigma|}^{\mathscr{D}})$, where D is a finite set, the set of* elements *of $\mathscr{D}$, and $R_i^{\mathscr{D}}$ is an $ar(R_i)$-ary relation on D. The set D is also called the* domain *of $\mathscr{D}$.*

We denote the maximum arity of the relation names in a schema σ as $\mathrm{ar}(\sigma)$. The *size* of a schema σ, denoted by $\|\sigma\|$, is the sum of the arities of its relation names.

We will often refer to relational databases as σ-dbs or databases (if the schema is not relevant). We assume that all databases $\mathscr{D}$ are linearly ordered or, equivalently, that $D = [n]$ for some $n \in \mathbb{N}$ (similar to [50]). We extend this linear ordering to a linear order on the relations of $\mathscr{D}$ via lexicographic ordering. We note that in this thesis we define the size of a database $\mathscr{D}$ to be $|D|$.

Remark 2.8. An undirected graph can be seen as a $\{E\}$-db, where E is a binary relation name, interpreted by a symmetric, irreflexive relation.

Let σ be a schema and let $\mathscr{D}$ be a σ-db. A *sub-database* of $\mathscr{D}$ is a σ-db that can be formed from $\mathscr{D}$ by removing a (possibly empty) subset of elements from $\mathscr{D}$ and a (possibly empty) subset of tuples from the relations of $\mathscr{D}$. Let $M \subseteq D$. The *induced sub-database* of $\mathscr{D}$ on M is the σ-db $\mathscr{D}[M]$ with domain M and $R^{\mathscr{D}[M]} := R^{\mathscr{D}} \cap M^{\mathrm{ar}(R)}$ for every $R \in \sigma$.

Let σ be a schema and let $\mathscr{D}$ and $\mathscr{D}'$ be σ-dbs. $\mathscr{D}$ and $\mathscr{D}'$ are *isomorphic* if there is a bijective map $f : D \to D'$ such that for every $R \in \sigma$ and $(a_1, \dots, a_{\mathrm{ar}(R)}) \in D^{\mathrm{ar}(R)}$, $(a_1, \dots, a_{\mathrm{ar}(R)}) \in R^{\mathscr{D}}$ if and only if $(f(a_1), \dots, f(a_{\mathrm{ar}(R)})) \in R^{\mathscr{D}'}$.

Throughout this thesis we will often refer to graph theoretic notions of relational databases (e.g. tree-width, connectedness, distance etc.). We use the *Gaifman graph* of a relational database to transfer graph notions to relational databases.

Definition 2.9 (Gaifman graph)**.** *Let σ be a schema. The* Gaifman graph *of a σ-db $\mathscr{D}$ is the undirected graph $\mathscr{G}(\mathscr{D}) = (D, E)$, where $ab \in E$ whenever $a \neq b$ and there is an $R \in \sigma$ and a tuple $(a_1, \dots, a_{ar(R)}) \in R^{\mathscr{D}}$ with $a, b \in \{a_1, \dots, a_{ar(R)}\}$.*

A database $\mathscr{D}$ is *connected* if the Gaifman graph $\mathscr{G}(\mathscr{D})$ is connected. The *connected components* of a database $\mathscr{D}$ are the maximal connected induced sub-databases of $\mathscr{D}$. For $k \in \mathbb{N}$ and schema σ, we fix a maximal set $\Psi(k, \sigma)$ of non-isomorphic connected σ-dbs on at most k elements.

We will now define multiple different classes of databases that will be used throughout this thesis. Note that we always assume that classes of databases are closed under isomorphism.

Definition 2.10 (Classes of databases with bounded degree)**.** *The* degree $\deg_{\mathscr{D}}(a)$ *of an element a in a database $\mathscr{D}$ is the total number of tuples in all relations of $\mathscr{D}$ that contain a. We say the* degree $\deg(\mathscr{D})$ *of a database $\mathscr{D}$ is the maximum degree of its elements.*

A class of databases $\mathbf{C}$ *has* bounded degree*, if there exists a constant $d \in \mathbb{N}$ such that for all $\mathscr{D} \in \mathbf{C}$, $\deg(\mathscr{D}) \leq d$.*

Let us remark that the $\deg(\mathscr{D})$ and the (graph-theoretic) degree of $\mathscr{G}(\mathscr{D})$ only differ by at most a constant factor (cf. e. g. [29]). Hence both measures yield the same classes of databases of bounded degree.

Definition 2.11 (Classes of databases with bounded tree-width)**.** *The* tree-width *of a database $\mathscr{D}$ is the the tree-width of its Gaifman graph.*

A class $\mathbf{C}$ *of databases has* bounded tree-width*, if there exists a constant $t \in \mathbb{N}$ such that all databases $\mathscr{D} \in \mathbf{C}$ have tree-width at most t.*

Let $\varepsilon \in [0,1]$, let $k \in \mathbb{N}$ and let σ be a schema. An (ε,k)*-partition* of a σ-db $\mathscr{D}$ on n elements is a σ-db $\mathscr{D}'$ formed by removing at most εn many tuples from $\mathscr{D}$ such that every connected component in $\mathscr{D}'$ contains at most k elements.

Definition 2.12 (Hyperfinite classes of databases)**.** *Let $\varepsilon \in [0,1]$ and let $k \in \mathbb{N}$. A database $\mathscr{D}$ is* (ε,k)-hyperfinite *if there exists an (ε,k)-partition of $\mathscr{D}$.*

Let $\rho : [0,1] \to \mathbb{R}^+$ be a function. A database $\mathscr{D}$ is ρ-hyperfinite *if for every $\varepsilon \in [0,1]$, $\mathscr{D}$ is $(\varepsilon,\rho(\varepsilon))$-hyperfinite. Let $\mathbf{C}'$ be a class of databases. The class $\mathbf{C} \subseteq \mathbf{C}'$ is* ρ-hyperfinite *on $\mathbf{C}'$ if for every $\varepsilon \in [0,1]$ and $\mathscr{D} \in \mathbf{C}$ there exists an $(\varepsilon,\rho(\varepsilon))$-partition $\mathscr{D}' \in \mathbf{C}'$ of $\mathscr{D}$. We call* $\mathbf{C}$ hyperfinite *on $\mathbf{C}'$ if there exists a function ρ such that $\mathbf{C}$ is ρ-hyperfinite on $\mathbf{C}'$.*

Definition 2.13 (Hereditary classes of databases)**.** *A class of databases is* hereditary *if it is closed under the removal of elements (i.e. it is closed under induced sub-databases).*

Let σ be a schema and let $\mathscr{D}$ and $\mathscr{B}$ be σ-dbs. Similar to graphs, we say $\mathscr{D}$ is *induced $\mathscr{B}$-free* if it contains no induced sub-database isomorphic to $\mathscr{B}$. For a family of databases F, we say $\mathscr{D}$ is *induced F-free* if it contains no induced sub-database isomorphic to any database in F. In Section 2.3 we noted that all hereditary graph classes have a (possibly infinite) set of forbidden induced subgraphs, this easily extends to databases. For a hereditary σ-db class $\mathbf{C}$, the *set of forbidden induced sub-databases F of* $\mathbf{C}$ is the minimal set of databases such that for any σ-db $\mathscr{D}$, $\mathscr{D} \in \mathbf{C}$ if and only if $\mathscr{D}$ is induced F-free. In other words, a σ-db $\mathscr{F}$ is in F if $\mathscr{F} \notin \mathbf{C}$, but any induced sub-database of $\mathscr{F}$ is in $\mathbf{C}$.

Definition 2.14 (Monotone classes of databases)**.** *A class of databases is* monotone *if it is closed under the removal of elements and tuples (i.e. it is closed under sub-databases).*

2.4.1 Neighbourhoods

For a database $\mathscr{D}$ and $a, b \in D$, the *distance* between a and b in $\mathscr{D}$, denoted by $\mathrm{dist}_{\mathscr{D}}(a,b)$, is the length of a shortest path between a and b in the Gaifman graph $\mathscr{G}(\mathscr{D})$ of $\mathscr{D}$. The *distance* between two tuples $\bar{a} = (a_1, \ldots, a_m)$ and $\bar{b} = (b_1, \ldots, b_\ell)$ of $\mathscr{D}$ is $\min\{\mathrm{dist}_{\mathscr{D}}(a_i, b_j) \mid 1 \leq i \leq m, 1 \leq j \leq \ell\}$.

Definition 2.15 (r-neighbourhood)**.** *Let $r \in \mathbb{N}$. For a tuple $\bar{a} \in D^{|\bar{a}|}$, we let $N_r^{\mathscr{D}}(\bar{a})$ denote the set of all elements of $\mathscr{D}$ that are at distance at most r from $\bar{a}$. The* r-neighbourhood *of $\bar{a}$ in $\mathscr{D}$, denoted by $\mathscr{N}_r^{\mathscr{D}}(\bar{a})$, is the tuple $(\mathscr{D}[N_r^{\mathscr{D}}(\bar{a})], \bar{a})$ where the elements of $\bar{a}$ are called* centres*. We omit the superscript and write $N_r(\bar{a})$ and $\mathscr{N}_r(\bar{a})$, if $\mathscr{D}$ is clear from the context.*

Let $d \geq 2$, let $r \in \mathbb{N}$ and let $\mathscr{D}$ be a database whose Gaifman graph has degree at most d. In [18] the authors show that $|N_r^{\mathscr{D}}(\bar{a})| \leq |\bar{a}| d^{r+1}$ where $\bar{a}$ is a tuple from $\mathscr{D}$. We can give a similar bound on databases of degree at most d using our definition of degree.

Lemma 2.16. *Let $d \geq 2$ and let $r \in \mathbb{N}$. Let σ be a schema and let $\mathscr{D}$ be a σ-db with degree at most d. For a tuple $\bar{a} = (a_1, \ldots, a_k)$ from $\mathscr{D}$,*

$$|N_r^{\mathscr{D}}(\bar{a})| \leq k(\mathrm{ar}(\sigma) \cdot d)^{r+1}.$$

Proof. The degree of the Gaifman graph of $\mathscr{D}$ will be at most $\mathrm{ar}(\sigma) \cdot d$. Therefore by Lemma 3.2 in [18], $|N_r^{\mathscr{D}}(\bar{a})| \leq k(\mathrm{ar}(\sigma) \cdot d)^{r+1}$. □

Two r-neighbourhoods, $(\mathscr{D}, \bar{a})$ and $(\mathscr{D}', \bar{b})$, are *isomorphic* if there is an isomorphism between $\mathscr{D}$ and $\mathscr{D}'$ which maps $\bar{a}$ to $\bar{b}$.

Definition 2.17 (r-neighbourhood type)**.** *Let $r \in \mathbb{N}$ and let $k \in \mathbb{N}_{\geq 1}$. An $\cong$-equivalence-class of r-neighbourhoods with k centres is called an* r-neighbourhood type *(or r-type for short) with k centres.*

Let $\mathscr{D}$ be a database. We say that tuple $\bar{a} \in D^{|\bar{a}|}$ has r-type τ*, if $\mathscr{N}_r^{\mathscr{D}}(\bar{a}) \in \tau$.*

We let $T_r^{\sigma,d}(k)$ denote the set of all r-types with k centres and degree at most d, over schema σ. Note that for fixed d and σ, the cardinality $|T_r^{\sigma,d}(k)| =: \mathrm{c}(r,k)$ is a constant, only depending on r and k. If $k = 1$, then we write $\mathrm{c}(r)$ instead of $\mathrm{c}(r,1)$ for short.

Definition 2.18 (Neighbourhood histogram and distribution vectors)**.** *Let $r \in \mathbb{N}$, let $k \in \mathbb{N}_{\geq 1}$ and let σ be a schema.*

- *The k-*centre r-histogram *of a σ-db $\mathscr{D}$, denoted by $\mathrm{h}_{r,k}(\mathscr{D})$, is the vector with $\mathrm{c}(r,k)$ components, indexed by the r-types in $T_r^{\sigma,d}(k)$, where the component corresponding to type τ contains the number of tuples in $\mathscr{D}$ of r-type τ.*

- *The* k-centre r-neighbourhood distribution *of* $\mathscr{D}$*, denoted by* $\mathrm{dv}_{r,k}(\mathscr{D})$*, is the vector* $\mathrm{h}_{r,k}(\mathscr{D})/n^k$ *where* $|D| = n$*.*
- *For a class of* σ*-dbs* $\mathbf{C}$*, we let* $\mathrm{h}_{r,k}(\mathbf{C}) := \{\mathrm{h}_{r,k}(\mathscr{D}) \mid \mathscr{D} \in \mathbf{C}\}$*.*

Note that if $k = 1$*, then we drop the subscript* k *and just write* $\mathrm{h}_r(\mathscr{D})$*,* $\mathrm{dv}_r(\mathscr{D})$ *and* $\mathrm{h}_r(\mathbf{C})$*.*

Lemma 5.1 in [57], allows approximating the one-centre r-neighbourhood distribution of an input graph of bounded degree by looking at a constant number of vertices. This result easily extends to bounded degree relational databases (e.g. see [4]). We can also easily extend this further to allow approximating the k-centre r-neighbourhood distribution of a bounded degree relational database for $k \in \mathbb{N}_{\geq 1}$ and $r \in \mathbb{N}$.

We let $\mathrm{EstimateFrequencies}_{r,k,s}$ be the algorithm, which given oracle access (see Section 2.6) to an input database $\mathscr{D}$ with degree at most $d \in \mathbb{N}$, proceeds as follows.

1. Let $\bar{v}$ be a vector with $c(r,k)$ components all of value 0.
2. Sample s tuples $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_s$ from D^k uniformly and independently.
3. For each sampled tuple $\bar{a}_i$, compute the r-type τ_j of $\bar{a}_i$. Set $\bar{v}[j] = \bar{v}[j] + 1/s$.
4. Return $\bar{v}$.

We assume that the r-types can be computed in constant time (we will discuss this in more detail in Section 2.6). $\mathrm{EstimateFrequencies}_{r,k,s}$ has constant running time, independent of $|D|$, and comes with the following guarantees (the proof of the following lemma is very similar to the proof of Lemma 5.1 in [57] but we give it for completeness). Recall that the ℓ_1*-norm* of a vector $\bar{v}$ on ℓ components is defined as $\|\bar{v}\|_1 := \sum_{i=1}^{\ell} |\bar{v}[i]|$.

Lemma 2.19. *Let* $\mathscr{D}$ *be a database on* n *elements with degree at most* $d \in \mathbb{N}$*. Let* $\lambda, \delta \in (0,1)$*, let* $k \in \mathbb{N}_{\geq 1}$*, and let* $r \in \mathbb{N}$*. If*

$$s \geq \frac{\mathrm{c}(r,k)^2}{\lambda^2} \cdot \ln\left(\frac{2\,\mathrm{c}(r,k)}{1-\delta}\right)$$

then with probability at least δ *the vector* $\bar{v}$ *returned by* $\mathrm{EstimateFrequencies}_{r,k,s}$ *on input* $\mathscr{D}$ *satisfies* $\|\bar{v} - \mathrm{dv}_{r,k}(\mathscr{D})\|_1 \leq \lambda$*.*

Proof. Let $X_{i,j}$ be the indicator random variable for the event that the r-type of $\bar{a}_i$ (the i-th tuple sampled in $\mathrm{EstimateFrequencies}_{r,k,s}$) is τ_j. Note that this event happens with probability $\mathrm{dv}_{r,k}(\mathscr{D})[j]$. Hence $\mathbb{E}[X_{i,j}] = \mathrm{dv}_{r,k}(\mathscr{D})[j]$ and $\mathbb{E}[\sum_{i=1}^{s} X_{i,j}] = s \cdot \mathrm{dv}_{r,k}(\mathscr{D})[j]$. Furthermore, we

have $\mathbb{E}[\bar{v}[j]] = \mathbb{E}[\sum_{i=1}^{s} X_{i,j}]/s = \mathrm{dv}_{r,k}(\mathscr{D})[j]$. For every $j \in [c(r,k)]$, using Chernoff bounds (see e.g., Theorem A.1.4 in [11]) we get,

$$\begin{aligned}
&\mathbb{P}\Big[|\bar{v}[j] - \mathrm{dv}_{r,k}(\mathscr{D})[j]| > \frac{\lambda}{c(r,k)}\Big] \\
&= \mathbb{P}\Big[\big|\sum_{i=1}^{s} X_{i,j} - \mathbb{E}[\sum_{i=1}^{s} X_{i,j}]\big| > \frac{\lambda s}{c(r,k)}\Big] \\
&\leq 2\exp\Big(\frac{-2\lambda^2 s}{c(r,k)^2}\Big) \\
&\leq 2\exp\Big(-2\ln\Big(\frac{2c(r,k)}{1-\delta}\Big)\Big) \\
&\leq \frac{1-\delta}{c(r,k)}.
\end{aligned}$$

Hence by the union bound the probability that for at least one $j \in [c(r,k)]$,

$$|\bar{v}[j] - \mathrm{dv}_{r,k}(\mathscr{D})[j]| > \frac{\lambda}{c(r,k)}$$

is at most $1-\delta$. Therefore with probability at least δ for all $j \in [c(r,k)]$,

$$|\bar{v}[j] - \mathrm{dv}_{r,k}(\mathscr{D})[j]| \leq \frac{\lambda}{c(r,k)}$$

and hence $\|\bar{v} - \mathrm{dv}_{r,k}(\mathscr{D})\|_1 \leq \lambda$ as required. □

Remark 2.20. The definitions of r-neighbourhoods, r-types, k-centre r-histograms and k-centre r-neighbourhood distributions directly translate to undirected graphs (since an undirected graph can be seen as a $\{E\}$-db where E is a binary relation name which is interpreted by a symmetric, irreflexive relation).

2.5 Logics and database queries

A *k-ary query* q is a computable function that maps a database $\mathscr{D}$ to a subset of D^k. If $k = 0$, then we call q a *boolean query*, and it maps $\mathscr{D}$ to either true or false.

In this thesis we will mainly be concerned with database queries that are definable in first-order logic (FO) and monadic second-order logic with counting (CMSO).

2.5.1 First-order logic

We will now introduce FO. For a more detailed introduction we direct the reader to the book [54]. We will start by defining FO formulas.

Definition 2.21 (First-order formulas)**.** *Let* ***var*** *be a countable infinite set of* variables *which will be typically denoted as x, y, z, ..., with subscripts and superscripts. Let us fix a relational schema* σ*. The set* $\mathrm{FO}[\sigma]$ *of* first-order formulas *over* σ *is inductively defined as follows.*

- *If* $x, y \in$ ***var****, then* $x = y$ *is an (atomic) formula.*
- *If* $R \in \sigma$ *and* $x_1, \ldots, x_{\mathrm{ar}(R)} \in$ ***var****, then* $R(x_1, \ldots, x_{\mathrm{ar}(R)})$ *is an (atomic) formula.*
- *If* ϕ_1 *and* ϕ_2 *are formulas, then* $\neg\phi_1$*,* $\phi_1 \vee \phi_2$*,* $\phi_1 \wedge \phi_2$*,* $\phi_1 \to \phi_2$ *and* $\phi_1 \leftrightarrow \phi_2$ *are formulas.*
- *If* ϕ *is a formula, then* $\exists x\phi$ *and* $\forall x\phi$ *(where* $x \in$ ***var****) are formulas.*

The set $\mathrm{FO}[\{E\}]$ is the set of first-order formulas for undirected graphs. We let $\mathrm{FO} := \bigcup_{\sigma \text{ schema}} \mathrm{FO}[\sigma]$. We use $\exists^{\geq m} x\,\phi$ (and $\exists^{=m} x\,\phi$, respectively) as a shortcut for the FO formula expressing that that the number of witnesses x satisfying ϕ is at least m (exactly m, resp.). The *quantifier rank* of a formula ϕ, denoted by $qr(\phi)$, is the maximum nesting depth of quantifiers that occur in ϕ. The *size of a formula* ϕ, denoted by $\|\phi\|$, is the length of ϕ as a string over the alphabet $\sigma \cup \mathbf{var} \cup \{\exists, \forall, \neg, \vee, \wedge, \to, \leftrightarrow, =\} \cup \{,\} \cup \{(,)\}$.

Definition 2.22 (Free variables)**.** *A* free variable *of an* FO *formula is a variable that does not appear in the scope of a quantifier. For a tuple* $\bar{x}$ *of variables and a formula* $\phi \in \mathrm{FO}$*, we write* $\phi(\bar{x})$ *to indicate that the free variables of* ϕ *are exactly the variables in* $\bar{x}$*.*

Definition 2.23 (Sentence)**.** *An* FO *formula without free variables is called a* sentence*.*

Let σ be a schema and let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with $|\bar{x}| = k$. Let $\mathscr{D}$ be a σ-db and $\bar{a}$ be a tuple of elements of $\mathscr{D}$ of length k. We write $\mathscr{D} \models \phi(\bar{a})$, if ϕ is true in $\mathscr{D}$ when we replace the free variables of ϕ with $\bar{a}$, and we say that $\bar{a}$ is an *answer* for ϕ in $\mathscr{D}$. We let $\phi(\mathscr{D}) := \{\bar{a} \in D^{|\bar{a}|} \mid \mathscr{D} \models \phi(\bar{a})\}$ be the set of all answers for ϕ on $\mathscr{D}$. If $k = 0$ (i.e. ϕ is a sentence) then we write $\mathscr{D} \models \phi$ to denote that ϕ is true in $\mathscr{D}$. Two formulas $\phi(\bar{x}), \psi(\bar{x}) \in \mathrm{FO}[\sigma]$, where $|\bar{x}| = k$, are *equivalent* (written $\phi(\bar{x}) \equiv \psi(\bar{x})$) if for all σ-dbs $\mathscr{D}$ and all $\bar{a} \in D^k$, $\mathscr{D} \models \phi(\bar{a})$ iff $\mathscr{D} \models \psi(\bar{a})$. Two formulas $\phi(\bar{x}), \psi(\bar{x}) \in \mathrm{FO}[\sigma]$, where $|\bar{x}| = k$, are *d-equivalent* (written $\phi(\bar{x}) \equiv_d \psi(\bar{x})$) if for all σ-dbs $\mathscr{D}$ with degree at most d and all $\bar{a} \in D^k$, $\mathscr{D} \models \phi(\bar{a})$ iff $\mathscr{D} \models \psi(\bar{a})$.

Hanf Normal Form

It is known that FO is local, in the sense that databases that locally look the same cannot be distinguished by FO formulas. This was formalised by Hanf [44] and Gaifman [37], and their results give rise to the normal forms for FO formulas called *Hanf normal form* and *Gaifman normal form*. We will introduce Hanf normal form only. Let us start by defining Sphere-formulas and Hanf-sentences.

Definition 2.24 (Sphere-formulas)**.** *Let $r \in \mathbb{N}$ and $k \in \mathbb{N}_{\geq 1}$. A* sphere-formula*, denoted by $\mathrm{sph}_\tau(\bar{x})$ (where $|\bar{x}| = k$), is an FO formula which expresses that the r-type of $\bar{x}$ is τ, where τ is some r-type with k centres, and r is called the* locality radius *of the sphere-formula.*

Definition 2.25 (Hanf-sentences)**.** *Let $r \in \mathbb{N}$ and $m \in \mathbb{N}$. A* Hanf-sentence *is a sentence of the form $\exists^{\geq m} x\, \mathrm{sph}_\tau(x)$, where τ is an r-type with one centre, and r is the* locality radius *of the Hanf-sentence.*

Definition 2.26 (Hanf normal form)**.** *An FO formula is in* Hanf normal form *if it is a Boolean combination of Hanf-sentences and sphere-formulas. The* Hanf locality radius *of an FO formula ϕ in Hanf normal form is the maximum of the locality radii of the Hanf-sentences and sphere-formulas of ϕ.*

A well-known theorem by Hanf states that on databases of bounded degree, every FO formula can be transformed into an equivalent formula in Hanf normal form [44]. This theorem was subsequently refined as follows.

Theorem 2.27 ([20])**.** *For any $\phi(\bar{x}) \in$ FO and $d \in \mathbb{N}_{\geq 1}$, there exists a d-equivalent formula $\psi(\bar{x})$ in Hanf normal form with the same free variables as ϕ. Furthermore, ψ can be computed in time $2^{d^{2^{\mathscr{O}(\|\phi\|)}}}$ from ϕ and the Hanf locality radius of ψ is at most $4^{qr(\phi)}$.*

2.5.2 Monadic second-order logic with counting

We will only briefly introduce CMSO (a detailed introduction can be found in [26]). *Monadic second-order logic* (MSO) is the extension of first-order logic that also allows quantification over subsets of the domain. CMSO extends MSO by allowing first-order modular counting quantifiers $\exists^m$ for every integer m (where $\exists^m \phi$ is true in a σ-db if the number of its elements for which ϕ is satisfied is divisible by m). A *free variable* of a CMSO formula is an individual or set variable that does not appear in the scope of a quantifier. As with FO, a formula without free variables is called a *sentence*. In this thesis we will not consider CMSO formulas with one or more free variables. For a σ-db $\mathscr{D}$ and a CMSO sentence ϕ (over σ) we write $\mathscr{D} \models \phi$ to denote that $\mathscr{D}$ satisfies ϕ.

2.6 Property testing

Throughout this thesis we will use the bounded degree database property testing model introduced in [4], which is a straightforward extension of the bounded degree graph model [42]. Hence in this section we will only introduce the bounded degree database model but for an in depth introduction to property testing we refer the reader to the book [39]. Note that throughout this thesis we will sometimes call the bounded degree database property testing model the BDRD model for short.

In this section we fix a schema σ, a degree bound $d \in \mathbb{N}$ and a class $\mathbf{C}$ of d-degree bounded σ-dbs.

In the bounded degree database model a *property* $\mathbf{P}$ on $\mathbf{C}$ is simply a subset of $\mathbf{C}$ which is closed under isomorphism. We say a σ-db $\mathscr{D}$ has the property $\mathbf{P}$ if and only if $\mathscr{D} \in \mathbf{P}$. Note that every FO and CMSO sentence (or more generally any boolean database query) ϕ defines a property $\mathbf{P}_\phi = \{\mathscr{D} \in \mathbf{C} \mid \mathscr{D} \models \phi\}$ on $\mathbf{C}$. We will often say that $\mathbf{P}_\phi$ is the property *defined by* ϕ *on* $\mathbf{C}$. The aim of a property testing algorithm for a property $\mathbf{P}$ on $\mathbf{C}$ is to decide with high probability correctly whether an input database from $\mathbf{C}$ (the *input class*) has $\mathbf{P}$ or is close to having $\mathbf{P}$ (i.e. it is a relaxation of the classical decision problem). Hence we need to define a distance measure between databases.

Definition 2.28 (Distance)**.** *Let $\mathscr{D}$ and $\mathscr{D}'$ be σ-dbs with degree at most d. The* distance *between $\mathscr{D}$ and $\mathscr{D}'$, denoted by* $\mathrm{dist}(\mathscr{D}, \mathscr{D}')$*, is the minimum number of tuples that have to be inserted or removed from relations of $\mathscr{D}$ and $\mathscr{D}'$ to make $\mathscr{D}$ and $\mathscr{D}'$ isomorphic. If it is not possible to make $\mathscr{D}$ and $\mathscr{D}'$ isomorphic by inserting or removing tuples (i.e. $|D| \neq |D'|$) then* $\mathrm{dist}(\mathscr{D}, \mathscr{D}') = \infty$.

Definition 2.29 (ε-close and ε-far)**.** *Let $\mathscr{D}$ and $\mathscr{D}'$ be σ-dbs with degree at most d that are both on n elements. For $\varepsilon \in [0,1]$, we say $\mathscr{D}$ and $\mathscr{D}'$ are* ε-close *if $\mathrm{dist}(\mathscr{D}, \mathscr{D}') \leq \varepsilon dn$, and are* ε-far *otherwise. Note that if $\mathscr{D}$ and $\mathscr{D}'$ are not on the same number of elements then they are ε-far.*

Let $\mathbf{P} \subseteq \mathbf{C}$ be some property. Then $\mathscr{D}$ is ε-close *to $\mathbf{P}$ if there exists a database $\mathscr{D}'' \in \mathbf{P}$ that is ε-close to $\mathscr{D}$, otherwise $\mathscr{D}$ is* ε-far *from $\mathbf{P}$.*

Property testing algorithms do not have access to the whole input. Instead, they are given access via an *oracle*. The type of queries allowed to the oracle depends on the property testing model (which in turn depends on the type and representation of the objects in the input class). In the bounded degree database model, the input class $\mathbf{C}$ is some class of d-bounded

degree σ-dbs. In this model a σ-db $\mathscr{D} \in \mathbf{C}$ on n elements is represented by a function

$$f_{\mathscr{D}} : \sigma \times [n] \times [d] \to \{\perp\} \cup \bigcup_{R \in \sigma} [n]^{\mathrm{ar}(R)}$$

where $f_{\mathscr{D}}(R,i,j)$ is the j^{th} tuple in $R^{\mathscr{D}}$ containing the i^{th} element[1] of D if one exists, otherwise $f_{\mathscr{D}}(R,i,j) = \perp$. In this model property testing algorithms are given $|D| = n$ as auxiliary input (we also assume the tester knows d and σ) and oracle access to $\mathscr{D}$ (the function $f_{\mathscr{D}}$). In particular in this model *oracle queries* are of the form (R,i,j) (where $R \in \sigma$, $i \leq n$ and $j \leq d$) and the answer to this query is $f_{\mathscr{D}}(R,i,j)$. We assume oracle queries are answered in constant time.

Remark 2.30. Given oracle access to a database $\mathscr{D}$ of degree at most $d \in \mathbb{N}$, for any $r \in \mathbb{N}$ the r-type of an element or tuple from $\mathscr{D}$ can be computed in time independent of $|D|$.

Let us now define an ε-tester.

Definition 2.31 (ε-tester)**.** *Let $\mathbf{P} \subseteq \mathbf{C}$ be a property and let $\varepsilon \in (0,1]$ be the proximity parameter. An* ε-tester *for $\mathbf{P}$ on $\mathbf{C}$ is a probabilistic algorithm which is given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}$ and it is given $n := |D|$ as auxiliary input. The algorithm does the following:*

1. *If $\mathscr{D} \in \mathbf{P}$, then the tester accepts with probability at least $2/3$.*
2. *If $\mathscr{D}$ is ε-far from $\mathbf{P}$, then the tester rejects with probability at least $2/3$.*

The ε-tester has one-sided error probability *if the tester accepts any $\mathscr{D} \in \mathbf{P}$ with probability* 1.

We note that the success probability $2/3$ of the ε-tester can be increased by repeatedly running the ε-tester and returning the majority outcome.

Definition 2.32 (Query complexity)**.** *Let $\mathbf{P} \subseteq \mathbf{C}$ be a property and let $\varepsilon \in (0,1]$. The* query complexity *of an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ is a function $q : \mathbb{N} \to \mathbb{N}$ where $q(n)$ is the maximum number of oracle queries made when the input has n elements. The ε-tester for $\mathbf{P}$ on $\mathbf{C}$ has* constant *query complexity if the query complexity does not depend on the size of the input database.*

We would like to point out here that the running time of an ε-tester can be much worse than its query complexity. The query complexity can be seen as a lower bound of the running time.

[1] According to the assumed linear order on D.

Definition 2.33 (Non-uniform testability)**.** *Let* $\mathbf{P} \subseteq \mathbf{C}$ *be a property. We say* $\mathbf{P}$ *is* non-uniformly testable *on* $\mathbf{C}$ *in time* $t(n)$*, if for every* $\varepsilon \in (0,1]$ *and* $n \in \mathbb{N}$ *there exists an* ε*-tester for the property* $\{\mathscr{D} \in \mathbf{P} \mid |D| = n\}$ *on the input class* $\{\mathscr{D} \in \mathbf{C} \mid |D| = n\}$ *which has constant query complexity and running time at most* $t(n)$*.*

Definition 2.34 (Uniform testability)**.** *Let* $\mathbf{P} \subseteq \mathbf{C}$ *be a property. We say* $\mathbf{P}$ *is* uniformly testable *on* $\mathbf{C}$ *in time* $t(n)$*, if for every* $\varepsilon \in (0,1]$ *there exists an* ε*-tester for* $\mathbf{P}$ *on* $\mathbf{C}$ *which has constant query complexity and whose running time on databases on n elements is at most* $t(n)$*. Note that this tester must work for all n.*

Adler and Harwath showed that, on the class of all databases with bounded degree and tree-width, every property definable in CMSO is uniformly testable in polylogarithmic running time [4]. (Where a function is *polylogarithmic* in n, if it is a polynomial in $\log n$.)

Theorem 2.35 ([4])**.** *Let* $t \in \mathbb{N}$ *and let* $\mathbf{C}_d^t$ *be the class of all d-degree bounded and t-bounded tree-width* σ*-dbs. Each property* $\mathbf{P} \subseteq \mathbf{C}_d^t$ *definable in* CMSO *is uniformly testable on* $\mathbf{C}_d^t$ *in polylogarithmic running time.*

Let us finish this section with an example.

Example 2.36. Let us take an e-commerce business that has a database containing information on all their products and sales. Let $\sigma = \{P,S\}$ be the schema where P and S are relation names and $\mathrm{ar}(P) = 3$ and $\mathrm{ar}(S) = 3$. In practice relation names also come with a set of *attributes*. In this example the attributes of P is the set {ProductId, Name, Price} and the attributes of S is the set {SaleId, ProductId, Date}. Now let us assume we want to query the database to find out whether all products have been sold at least once. This query is definable by the FO sentence

$$\phi := \forall x \forall y_1 \forall y_2 \exists z_1 \exists z_2 (P(x,y_1,y_2) \to S(z_1,x,z_2)).$$

Let $\mathbf{C}$ be the class of all σ-dbs with degree at most d. Let $\mathbf{P} \subseteq \mathbf{C}$ be the property defined by ϕ on $\mathbf{C}$. Then we claim that $\mathbf{P}$ is uniformly testable on $\mathbf{C}$ in constant time. Let $\varepsilon \in (0,1]$. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}$ on n elements, the ε-tester proceeds as follows.

1. Uniformly and independently sample $\alpha = log_{1-\varepsilon}\frac{1}{3}$ elements from $[n]$.
2. For each of the sampled elements i do the following.
   (a) Perform the oracle queries $(P,i,1),\ldots,(P,i,d)$.
   (b) If a tuple is returned from one of these queries with i in the first component perform the oracle queries $(S,i,1),\ldots,(S,i,d)$.

3. If in Step 2 an element, i, was found such that there is a tuple in the relation $P^{\mathscr{D}}$ with i in the first component but no tuple in the relation $S^{\mathscr{D}}$ with i in the second component, then reject. Otherwise, accept.

Clearly, this tester has constant query complexity and constant running time.

Let us now prove correctness. Let $\mathscr{D} \in \mathbf{P}$. Then clearly the tester will always accept.

Now let $\mathscr{D}$ be ε-far from $\mathbf{P}$. Therefore we need to insert or remove more than εdn tuples to make $\mathscr{D}$ have the property. Let us call an element $i \in [n]$ *bad* if there is a tuple in the relation $P^{\mathscr{D}}$ with i in the first component but no tuple in the relation $S^{\mathscr{D}}$ with i in the second component. For every bad $i \in [n]$ we can simply remove all tuples from $P^{\mathscr{D}}$ which contain i. This requires at most d tuple modifications. Therefore there are at least εn many distinct bad $i \in [n]$. The probability that a uniformly sampled element from $[n]$ is bad is therefore at least ε. The probability that we sample no bad elements is at most $(1-\varepsilon)^{\alpha} = \frac{1}{3}$. Hence the probability the tester rejects is at least $\frac{2}{3}$ as required.

2.7 Semilinear sets

In Chapters 4 and 5 we study properties whose set of r-histogram vectors form a *semilinear set*.

Definition 2.37 (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*

$$M = \{\bar{v}_0 + a_1\bar{v}_1 + \cdots + a_k\bar{v}_k \mid a_1, \ldots, a_k \in \mathbb{N}\},$$

for some $\bar{v}_0, \ldots, \bar{v}_k \in \mathbb{N}^c$.

We can show that the set of r-histograms of the class of all d-degree bounded σ-dbs whose connected components are all of size at most k is linear.

Lemma 2.38. *Let $k, r, d \in \mathbb{N}$ and let σ be a schema. Let $\mathbf{C}$ be the class of all σ-dbs of degree at most d, whose connected components are all of size at most k. Then $\mathrm{h}_r(\mathbf{C})$ is linear.*

Proof. Recall $\Psi(k,\sigma)$ is a maximal set of non-isomorphic connected σ-dbs of size at most k. Let $\Psi := \Psi(k,\sigma) = \{\mathscr{D}_1, \ldots, \mathscr{D}_{|\Psi|}\}$. Let

$$M = \{a_1 \mathrm{h}_r(\mathscr{D}_1) + \cdots + a_{|\Psi|}\mathrm{h}_r(\mathscr{D}_{|\Psi|}) \mid a_1, \ldots, a_{|\Psi|} \in \mathbb{N}\}.$$

Claim 2.39. $\mathrm{h}_r(\mathbf{C}) = M$.

Proof: Let $\mathscr{D} \in \mathbf{C}$. Then there exists $a_1, \ldots, a_{|\Psi|} \in \mathbb{N}$ such that $\mathscr{D}$ contains a_1 connected components isomorphic to $\mathscr{D}_1$, a_2 connected components isomorphic to $\mathscr{D}_2$, ..., and $a_{|\Psi|}$ connected components isomorphic to $\mathscr{D}_{|\Psi|}$. Therefore

$$\mathrm{h}_r(\mathscr{D}) = a_1\,\mathrm{h}_r(\mathscr{D}_1) + \cdots + a_{|\Psi|}\,\mathrm{h}_r(\mathscr{D}_{|\Psi|}) \in M.$$

Let $\bar{m} = a_1\,\mathrm{h}_r(\mathscr{D}_1) + \cdots + a_{|\Psi|}\,\mathrm{h}_r(\mathscr{D}_{|\Psi|}) \in M$ where $a_1, \ldots, a_{|\Psi|} \in \mathbb{N}$. Then the σ-db $\mathscr{D}$ with exactly a_1 connected components isomorphic to $\mathscr{D}_1$, a_2 connected components isomorphic to $\mathscr{D}_2$, ..., and $a_{|\Psi|}$ connected components isomorphic to $\mathscr{D}_{|\Psi|}$ is in $\mathbf{C}$ and clearly $\mathrm{h}_r(\mathscr{D}) = \bar{m}$. Hence $\mathrm{h}_r(\mathbf{C}) = M$. ■

Therefore $\mathrm{h}_r(\mathbf{C})$ is a linear set. □

From a result in [34] about many-sorted spectra of CMSO sentences it can be derived that the set of r-histogram vectors of properties defined by a CMSO sentence on the class of all bounded degree and bounded tree-width databases are semilinear.

Lemma 2.40 ([4, 34])**.** *Let $t, d \in \mathbb{N}$, let σ be a schema and let $\mathbf{C}^t_d$ be the class of all d-degree bounded and t-bounded tree-width σ-dbs. For each $r \in \mathbb{N}$ and each property $\mathbf{P} \subseteq \mathbf{C}^t_d$ definable by a CMSO sentence on $\mathbf{C}^t_d$, the set $\mathrm{h}_r(\mathbf{P})$ is semilinear.*

Chapter 3

Testing first-order definable properties

In this chapter, we consider the testability of properties definable in two different fragments of FO in the bounded degree database model. We first consider the *existential fragment of FO* in Section 3.1. A FO sentence is in the existential fragment of FO if it is of the form $\exists x_1 \ldots \exists x_\ell \psi(x_1, \ldots, x_\ell)$ where ψ is quantifier free. Given a FO existential sentence ϕ and database $\mathscr{D}$ it is not difficult to see that we only need to modify a constant number of tuples in $\mathscr{D}$ to make it satisfy ϕ. Therefore if $\mathscr{D}$ is large enough then it is close to the property defined by ϕ and hence properties definable in the existential fragment of FO are trivially uniformly testable in the bounded degree database model. We note here that *boolean conjunctive queries* (FO sentences constructed from atomic formulas, conjunctions and existential quantifiers only), which are one of the most studied class of boolean database queries, belong to the existential fragment of FO. Hence boolean conjunctive queries are trivially uniformly testable in the bounded degree database model. A natural question one may ask is whether properties definable by the negation of a boolean conjunctive query ϕ are uniformly testable? The negation of a boolean conjunctive query belongs to the *universal fragment of FO*. We will consider the testability of the universal fragment of FO in Section 3.2. A FO sentence is in the universal fragment of FO if it is of the form $\forall x_1 \ldots \forall x_\ell \psi(x_1, \ldots, x_\ell)$ where ψ is quantifier free. A sentence of the form $\forall x_1 \ldots \forall x_\ell \psi(x_1, \ldots, x_\ell)$ is logically equivalent to the sentence $\neg \exists x_1 \ldots \exists x_\ell \neg \psi(x_1, \ldots, x_\ell)$. Therefore it is easy to show that testing a property defined by a universal FO sentence is equivalent to testing induced B-freeness where B is some finite set of databases. In Section 3.2 we prove that the property of being induced B-free is uniformly testable in constant time in the bounded degree database model (Theorem 3.8) and hence properties definable in the universal fragment of FO are uniformly testable in constant time (Theorem 3.3).

Proviso. *In this chapter we fix a schema σ and number $d \in \mathbb{N}$ with $d \geq 2$. All databases are σ-dbs and have degree at most d, unless stated otherwise. We use $\mathbf{C}_d$ to denote the class of all σ-dbs with degree at most d.*

3.1 Existential fragment

It is easy to see that properties defined by an existential FO sentence are trivially uniformly testable in the bounded degree database model. Let us start with a simple example of a property defined by an existential FO sentence.

Example 3.1. Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property defined by the FO sentence

$$\phi = \exists x_1 \ldots \exists x_r R(x_1, ..., x_r)$$

where $R \in \sigma$. We can show that $\mathbf{P}$ is uniformly testable on $\mathbf{C}_d$ with constant time complexity.

Let $\varepsilon \in (0,1]$, given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. If $n < \frac{1}{\varepsilon d}$ then do a full check for a tuple in R.
2. Otherwise, accept.

Clearly, the tester has constant query complexity and constant running time.

For correctness, if $\mathscr{D} \in \mathbf{P}$ then the tester will always accept.

Let $\mathscr{D}$ be ε-far from $\mathbf{P}$. We only need to insert one tuple into $\mathscr{D}$ to make $\mathscr{D} \in \mathbf{P}$. Hence we must have $1 > \varepsilon dn$, in which case the tester does a full check for a tuple in R and hence will reject.

We will give the proof of the following observation for completeness.

Observation 3.2. Every property $\mathbf{P} \subseteq \mathbf{C}_d$ that is definable by an existential FO sentence is (trivially) uniformly testable on $\mathbf{C}_d$ with constant time complexity.

Proof. Let ϕ be an existential FO sentence and let $\mathbf{P}$ be the property defined by ϕ on $\mathbf{C}_d$. We will assume that $\mathbf{P}$ is non-empty (otherwise a tester can just reject). Let $\varepsilon \in (0,1]$ and let c be the minimum number such that for every $\mathscr{D} \in \mathbf{C}_d$, $\mathscr{D}$ can be modified into a database $\mathscr{D}'$ (with degree at most d) with at most c tuple modifications such that $\mathscr{D}' \models \phi$. We note that c is bounded above by the number of existential quantifiers in ϕ and d and hence is a constant. Then the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ is very similar to the tester given in Example 3.1. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. If $n < \frac{c}{\varepsilon d}$, then decide exactly if $\mathscr{D}$ is in $\mathbf{P}$ (check whether $\mathscr{D} \models \phi$).

2. Otherwise, accept.

Clearly, the tester has constant query complexity and constant running time.

The proof of correctness is then very similar to the proof of correctness given in Example 3.1. □

3.2 Universal fragment

The existential fragment of FO logic was trivial and not very interesting, we will therefore move our attention to the not so trivial universal fragment of FO logic. In this section, we will prove that any property that is definable by a universal FO sentence is uniformly testable on $\mathbf{C}_d$ in constant time, which to our knowledge has not been done before in the bounded degree database model.

Theorem 3.3. *Every property $\mathbf{P} \subseteq \mathbf{C}_d$ that is definable by a universal FO sentence is uniformly testable on $\mathbf{C}_d$ with constant time complexity.*

As with the existential fragment let us start with a simple example.

Example 3.4. Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property defined by the FO sentence

$$\phi = \forall x_1 \ldots \forall x_r \neg R(x_1, \ldots, x_r)$$

where $R \in \sigma$ is a relation symbol and $r \in \mathbb{N}$. We will show that $\mathbf{P}$ is uniformly testable on $\mathbf{C}_d$ with constant time complexity.

Let $\varepsilon \in (0,1]$. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, a ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. Sample $\alpha = log_{1-\varepsilon}\frac{1}{3}$ elements from $[n]$ uniformly and independently.

2. For each of the sampled elements, i, perform the query $(R, i, 1)$.

3. If in the previous step all queries returned no results then accept, otherwise reject.

If $\mathscr{D} \in \mathbf{P}$ (there are no tuples in relation $R^{\mathscr{D}}$) then clearly the ε-tester will always accept.

Let $\mathscr{D}$ be ε-far from property $\mathbf{P}$. Therefore there must be more than εdn tuples in the relation $R^{\mathscr{D}}$. As the maximum degree of $\mathscr{D}$ is d there must be at least εn distinct elements in the tuples of the relation $R^{\mathscr{D}}$. The probability of sampling one of these elements is

therefore at least $\varepsilon n/n = \varepsilon$. The probability of not choosing one of these elements is at most $(1-\varepsilon)^{\alpha} = 1/3$ by the choice of α. Hence the ε-tester rejects with probability of at least $2/3$.

This tester has constant running time and query complexity. It also has one sided error probability.

Any universal FO sentence is logically equivalent to a sentence of the form

$$\phi = \neg\exists x_1 \exists x_2 \ldots \exists x_\ell \psi(x_1, x_2, ..., x_\ell)$$

where ψ is quantifier free and $\ell \in \mathbb{N}$. Hence any property definable by a universal FO sentence is hereditary. It is known that all hereditary graph properties have a (possibly infinite) set of forbidden induced subgraphs (e.g. see [10]) and this easily extends to databases. We can show that properties definable by a universal FO sentence have a finite set of forbidden induced sub-databases.

Lemma 3.5. *Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property defined by the sentence*

$$\phi = \forall x_1 \forall x_2 \ldots \forall x_\ell \psi(x_1, x_2, \ldots, x_\ell)$$

where ψ is quantifier free and $\ell \in \mathbb{N}$. Then $\mathbf{P}$ has a finite set $B = \{\mathscr{B}_1, \mathscr{B}_2, ..., \mathscr{B}_m\} \subseteq \mathbf{C}_d$ of forbidden induced sub-databases. Furthermore, such a set B is computable.

Proof. Let us construct the set B as follows.

1. Let $B = \emptyset$.
2. For $1 \leq i \leq \ell$ do the following.
   (a) Let Π_i be the set of all σ-dbs in $\mathbf{C}_d$ on i elements.
   (b) For each $\mathscr{B} \in \Pi_i$, if $\mathscr{B} \models \neg\phi$ and B contains no σ-db that is an induced sub-database of $\mathscr{B}$, then add $\mathscr{B}$ to B.

Let $\Pi = \bigcup_{1 \leq i \leq \ell} \Pi_i$. For every $\mathscr{B} \in \Pi$ it can be checked whether $\mathscr{B} \models \neg\phi$ in time only dependent on ϕ, d and σ (see [60] and [36]). Furthermore, the set Π is finite ($|\Pi|$ depends only on σ, d and ℓ) and hence B is finite and computable (in time only dependent on ϕ, d and σ).

Let us now show that B is indeed a set of forbidden induced sub-databases of $\mathbf{P}$. Let $\mathscr{D} \in \mathbf{C}_d$. We will first show that if $\mathscr{D}$ is induced B-free then $\mathscr{D} \in \mathbf{P}$. For a contradiction let us assume that $\mathscr{D}$ is induced B-free but $\mathscr{D} \notin \mathbf{P}$. Since $\mathscr{D} \notin \mathbf{P}$, then $\mathscr{D} \not\models \phi$ and hence $\mathscr{D} \models \neg\phi = \exists x_1 \exists x_2 \ldots \exists x_\ell \neg\psi(x_1, x_2, \ldots, x_\ell)$. Therefore there exists some tuple of elements

$\bar{a} = (a_1, a_2, \ldots, a_\ell) \in D^\ell$ such that $\neg\psi(a_1, a_2, \ldots, a_\ell)$ is true in $\mathscr{D}$. Let $\mathscr{D}'$ be the induced sub-database of $\mathscr{D}$ on the elements in $\bar{a}$. Then $\mathscr{D}'$ has at most ℓ elements and hence $\mathscr{D}' \in \Pi$. Furthermore, $\mathscr{D}' \models \neg\phi = \exists x_1 \exists x_2 \ldots \exists x_\ell \neg\psi(x_1, x_2, \ldots, x_\ell)$. Therefore either $\mathscr{D}'$ or an induced sub-database of $\mathscr{D}'$ will be in B. Hence $\mathscr{D}$ is not induced B-free, which is a contradiction. Therefore if $\mathscr{D}$ is induced B-free then $\mathscr{D} \in \mathbf{P}$.

Now let us show that if $\mathscr{D} \in \mathbf{P}$ then $\mathscr{D}$ is induced B-free. For a contradiction let us assume that $\mathscr{D} \in \mathbf{P}$ but $\mathscr{D}$ is not induced B-free. Hence there exists at least one $\mathscr{B}_i \in B$ such that $\mathscr{D}$ has an induced sub-database isomorphic to $\mathscr{B}_i$. Since $\mathscr{B}_i \models \exists x_1 \exists x_2 \ldots \exists x_\ell \neg\psi(x_1, x_2, \ldots, x_\ell)$ and $\mathscr{D}$ has an induced sub-database isomorphic to $\mathscr{B}_i$, $\mathscr{D}$ must satisfy $\exists x_1 \exists x_2 \ldots \exists x_\ell \neg\psi(x_1, x_2, \ldots, x_\ell)$. This contradicts that $\mathscr{D} \in \mathbf{P}$ and hence if $\mathscr{D} \in \mathbf{P}$ then $\mathscr{D}$ is induced B-free.

Finally, by the construction of B, B is minimal and therefore B is the set of forbidden induced sub-databases of $\mathbf{P}$. □

To prove Theorem 3.3 we need to show that induced B-freeness (where B is some finite set of σ-dbs) is uniformly testable in constant time. We will start by proving that induced $\mathscr{B}$-freeness (where $\mathscr{B}$ is a connected σ-db) is uniformly testable in constant time and then extend this to cases where $\mathscr{B}$ is disconnected.

Goldreich and Ron [42] proved that on bounded degree graphs, the property of being (not necessarily induced) $\mathscr{H}$-free, where $\mathscr{H}$ is a connected graph, is uniformly testable in constant time. It was left as an exercise in [39] to prove this for the case when $\mathscr{H}$ is disconnected. We will complete this exercise, modify the testers to test for induced $\mathscr{H}$-free properties and translate the proofs from graphs to databases.

Theorem 3.6. *Let $\mathscr{B}$ be a connected σ-db. Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property containing all induced $\mathscr{B}$-free σ-dbs in $\mathbf{C}_d$. Then $\mathbf{P}$ is uniformly testable on $\mathbf{C}_d$ with constant time complexity and one sided error probability.*

Proof. Let r be the largest distance between any two elements in $\mathscr{B}$ and let $\varepsilon \in (0,1]$. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. Uniformly and independently sample $\alpha = log_{1-\varepsilon} 1/3$ elements from $[n]$.
2. Explore the r-neighbourhood around each sampled element.
3. If the r-neighbourhood of any of the sampled elements contains an induced sub-database isomorphic to $\mathscr{B}$, then reject. Otherwise, accept.

The r-neighbourhood of an element can be explored in constant time and with constant query complexity. It can be checked in constant time whether the r-neighbourhood of an

element has an induced sub-database isomorphic to $\mathscr{B}$. Finally, since we sample a constant number of elements, the tester has constant query complexity and constant running time. Let us now prove correctness.

If $\mathscr{D} \in \mathbf{P}$ then the tester will clearly always accept and hence the tester has one sided error probability.

Now let us assume that $\mathscr{D}$ is ε-far from $\mathbf{P}$. Therefore we need to insert/remove more than εdn tuples to remove all copies of $\mathscr{B}$ from $\mathscr{D}$. Let $D_B \subseteq D$ be the set of elements in $\mathscr{D}$ that belong to an induced sub-database of $\mathscr{D}$ which is isomorphic to $\mathscr{B}$. We claim that $|D_B| > \varepsilon n$. We will prove that $|D_B| > \varepsilon n$ by a contradiction. Let us assume that $|D_B| \leq \varepsilon n$. For each element $b \in D_B$, if $|B| > 1$, then remove all tuples from $\mathscr{D}$ that contain b (a maximum of d tuples), otherwise either insert or remove a tuple that only contains the element b. Let $\mathscr{D}'$ be the resulting database. If $|B| > 1$, then by construction any induced sub-database of $\mathscr{D}'$ on $|B|$ elements that contains some element in D_B will now have at least two connected components. If $|B| = 1$, then by construction any induced sub-database of $\mathscr{D}'$ on an element of D_B will not be isomorphic to $\mathscr{B}$. Hence the resulting database $\mathscr{D}'$ does not have an induced sub-database isomorphic to $\mathscr{B}$. Therefore if $|D_B| \leq \varepsilon n$, then $\mathscr{D}$ must be ε-close to $\mathbf{P}$ which is a contradiction. Therefore $|D_B| > \varepsilon n$. By the choice of r if the tester samples any element in D_B, an induced sub-database isomorphic to $\mathscr{B}$ will be found and the tester will reject. The probability an element in D_B is sampled is $|D_B|/n > \varepsilon$. The probability that out of the α elements sampled, no element in D_B is sampled is at most $(1-\varepsilon)^{\alpha} = 1/3$. Hence the probability the tester rejects is at least $2/3$. □

Theorem 3.7. *Let $\mathscr{B}$ be a disconnected σ-db. Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property containing all induced $\mathscr{B}$-free σ-dbs in $\mathbf{C}_d$. Then $\mathbf{P}$ is uniformly testable on $\mathbf{C}_d$ with constant time complexity and one sided error probability.*

Proof. Let $\varepsilon \in (0,1]$ and let $\mathscr{B}_1, \mathscr{B}_2, \ldots, \mathscr{B}_k$ be the connected components of $\mathscr{B}$. For $i \in [k]$, let r_i be the largest distance between any two elements in $\mathscr{B}_i$. Let $r_{max} = \max\{r_1, \ldots, r_k\}$. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. If $n < \frac{2k(\mathrm{ar}(\sigma)\cdot d)^{2r_{\max}+2}}{\varepsilon}$ then decide exactly if $\mathscr{D} \in \mathbf{P}$.
2. Uniformly and independently sample $\alpha = \log_{1-(\varepsilon/2)^k} 1/3$ many tuples from $[n]^k$.
3. For each sampled tuple $\bar{b}$, check if $\mathscr{D}[N_{r_{\max}}(\bar{b})]$ contains an induced sub-database isomorphic to $\mathscr{B}$, and if so reject.
4. If no induced sub-database isomorphic to $\mathscr{B}$ was found in the previous step, then accept.

If $n < 2k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}/\varepsilon$, then it can be checked in constant time and with constant query complexity if $\mathscr{D} \in \mathbf{P}$ (since $2k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}/\varepsilon$ is a constant). Otherwise, for each sampled tuple $\bar{b}$, $\mathscr{D}[N_{r_{\max}}(\bar{b})]$ can be computed in constant time and with constant query complexity. Furthermore, it can be checked in constant time whether $\mathscr{D}[N_{r_{\max}}(\bar{b})]$ has an induced sub-database isomorphic to $\mathscr{B}$. Finally, since a constant number of tuples are sampled the overall running time and query complexity is constant as required.

If $\mathscr{D} \in \mathbf{P}$ then clearly the tester will always accept. Hence the tester has one sided error probability.

Let $\mathscr{D}$ be ε-far from $\mathbf{P}$. Let us assume that $n \geq 2k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}/\varepsilon$ (otherwise the tester will reject with probability 1). Firstly, let us note that $\mathscr{D}$ must be ε-far from being $\mathscr{B}_i$-free for all $i \in [k]$ (otherwise $\mathscr{D}$ would be ε-close to $\mathbf{P}$). For each $i \in [k]$, let D_{B_i} be the set of elements in $\mathscr{D}$ that belong to an induced sub-database of $\mathscr{D}$ which is isomorphic to $\mathscr{B}_i$. With the same arguments as in the proof of Theorem 3.6 it can be deduced that $|D_{B_i}| > \varepsilon n$ for every $i \in [k]$. Now let us find a lower bound on the number of tuples in $[n]^k$ which will lead to an induced sub-database isomorphic to $\mathscr{B}$ being found in Step 3. For ease let us compute a lower bound on the number of tuples $(b_1,\ldots,b_k) \in [n]^k$ where $b_i \in D_{B_i}$ for every $i \in [k]$ and for every $i, j \in [k]$ with $i \neq j$ we have that b_i and b_j are at distance at least $2r_{\max}+2$ away from each other in $\mathscr{D}$, i.e. $\mathscr{D}[N_{r_{\max}}(b_i) \cup N_{r_{\max}}(b_j)]$ is disconnected (note that such tuples will lead to an induced sub-database isomorphic to $\mathscr{B}$ being found in Step 3). There are at least $(\varepsilon n - k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2})^k$ many such tuples in $[n]^k$ since for each $i \in [k]$, $|D_{B_i}| > \varepsilon n$, and by Lemma 2.16 there are at most $(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}$ many elements in the $2r_{\max}+1$-neighbourhood of an element in $\mathscr{D}$. The probability a tuple is sampled that leads to an induced sub-database isomorphic to $\mathscr{B}$ being found in Step 3 is therefore at least

$$\left(\frac{\varepsilon n - k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}}{n}\right)^k \geq \left(\frac{\varepsilon}{2}\right)^k$$

since $n \geq 2k(\text{ar}(\sigma)\cdot d)^{2r_{\max}+2}/\varepsilon$. The probability we don't find an induced sub-database isomorphic to $\mathscr{B}$ is then at most

$$\left(1 - \left(\frac{\varepsilon}{2}\right)^k\right)^{\alpha} = \frac{1}{3}$$

and hence the probability the tester rejects is at least $2/3$. □

To prove Theorem 3.3 it now only remains to prove that the intersection of induced $\mathscr{B}$-free properties are also uniformly testable in constant time. Goldreich and Ron [42] prove that in the bounded degree graph model the intersection of uniformly testable (decreasing) monotone graph properties are also uniformly testable. Therefore the intersection of (not

necessarily induced) subgraph-free properties are uniformly testable. Their proof requires that the properties are closed under edge removal and hence we can not use their result.

Theorem 3.8. *Let $B = \{\mathscr{B}_1, \mathscr{B}_2, \ldots, \mathscr{B}_m\}$ be a finite set of σ-dbs. Let $\mathbf{P} \subseteq \mathbf{C}_d$ be the property containing all induced B-free σ-dbs in $\mathbf{C}_d$. Then $\mathbf{P}$ is uniformly testable on $\mathbf{C}_d$ with constant time complexity.*

Proof. Let c be the number of different 1-element σ-dbs up to isomorphism (note c is a constant as it only depends on d and σ). Let $B_{\max} = \max\{|B_1|, |B_2|, \ldots, |B_m|\}$, let $\varepsilon \in (0,1]$, and let $\varepsilon' = \frac{\varepsilon}{2dmc}$. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d$ on n elements, a ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$ proceeds as follows.

1. If $n < 2cB_{\max}(\mathrm{ar}(\sigma)\cdot d)^2$, then decide exactly if $\mathscr{D} \in \mathbf{P}$.
2. For each $\mathscr{B}_i \in B$ do the following.
   (a) Run the ε'-tester for induced $\mathscr{B}_i$-freeness from Theorem 3.6 (if $\mathscr{B}_i$ is connected) or from Theorem 3.7 (if $\mathscr{B}_i$ is disconnected) on $\mathscr{D}$.
   (b) If the tester in (a) rejected, then reject.
3. If all the testers run in Step 2 accepted, then accept.

Since $|B|$ is a constant and the ε'-testers from the proofs of Theorem 3.6 and Theorem 3.7 run in constant time and have constant query complexity, the tester has constant query complexity and constant running time as required.

If $\mathscr{D} \in \mathbf{P}$ then the tester will always accept (since the ε'-testers from the proofs of Theorem 3.6 and Theorem 3.7 have one sided error probability). Hence the tester has one sided error probability.

Let $\mathscr{D}$ be ε-far from $\mathbf{P}$. Let us assume that $n \geq 2cB_{\max}(\mathrm{ar}(\sigma)\cdot d)^2$ (otherwise the tester will reject with probability 1). We need to show that for at least one $i \in [m]$, $\mathscr{D}$ is ε'-far from being induced $\mathscr{B}_i$-free and hence the tester will reject with probability at least $2/3$. For a contradiction let us assume that $\mathscr{D}$ is ε'-close to being induced $\mathscr{B}_i$-free for all $i \in [m]$. For $i \in [m]$, let $\mathscr{D}_i \in \mathbf{C}_d$ be a σ-db that is induced $\mathscr{B}_i$-free and is formed by adding/removing at most $\varepsilon' nd$ tuples to/from $\mathscr{D}$. Let $\mathscr{D}'$ be the maximal induced sub-database of $\mathscr{D}$, $\mathscr{D}_1$, $\mathscr{D}_2$, …, and $\mathscr{D}_m$. Clearly, $\mathscr{D}' \in \mathbf{P}$ but $\mathscr{D}'$ might have fewer elements than $\mathscr{D}$. We will show that we can add $n - |D'|$ isolated elements to $\mathscr{D}'$ in such a way that the resulting database $\mathscr{D}''$ is still in $\mathbf{P}$ and is ε-close to $\mathscr{D}$, which will give us a contradiction.

By the pigeon hole principal, there exists some σ-db $\mathscr{H}$ on one element such that for at least n/c elements $b \in D$, $\mathscr{D}[b] \cong \mathscr{H}$. We will show that if we insert $n - |D'|$ disjoint copies

of $\mathscr{H}$ into $\mathscr{D}'$ then the resulting database will still be induced $\mathscr{B}_i$-free for all $\mathscr{B}_i \in B$. Let $\mathscr{B}_i \in B$. We will consider the three cases where, (1) $\mathscr{B}_i$ contains no connected component isomorphic to $\mathscr{H}$, (2) every connected component in $\mathscr{B}_i$ is isomorphic to $\mathscr{H}$, and (3) $\mathscr{B}_i$ contains at least one connected component isomorphic to $\mathscr{H}$ and at least one connected component not isomorphic to $\mathscr{H}$.

For case (1), clearly, no matter how many disjoint copies of $\mathscr{H}$ are inserted into $\mathscr{D}'$, the resulting database will still be induced $\mathscr{B}_i$-free.

For case (2), let us assume that every connected component in $\mathscr{B}_i$ is isomorphic to $\mathscr{H}$. We will show that if this is the case then $\mathscr{D}$ must be ε'-far from being induced $\mathscr{B}_i$-free which contradicts our earlier assumption (and hence case (2) is not possible). If there exists $|B_i|$ distinct elements $b \in D$ such that $\mathscr{D}[b] \cong \mathscr{H}$ and such that no two are at distance 1 from each other, then $\mathscr{D}$ will have an induced sub-database isomorphic to $\mathscr{B}_i$. If there are at least $|B_i|(\mathrm{ar}(\sigma)\cdot d)^2$ distinct elements $b \in D$ such that $\mathscr{D}[b] \cong \mathscr{H}$, then at least $|B_i|$ many of them will all be at distance greater than 1 from each other (since there are at most $(\mathrm{ar}(\sigma)\cdot d)^2$ elements in the 1-neighbourhood of an element by Lemma 2.16). Since there are at least n/c elements $b \in D$ where $\mathscr{D}[b] \cong \mathscr{H}$, to make $\mathscr{D}$ be induced $\mathscr{B}_i$-free we will need to insert or remove at least

$$\frac{n}{c} - |B_i|(\mathrm{ar}(\sigma)\cdot d)^2 \geq \frac{n}{c} - B_{\max}(\mathrm{ar}(\sigma)\cdot d)^2 \geq \frac{n}{2c} > \frac{\varepsilon n}{2mc} = \varepsilon' nd$$

tuples since $n \geq 2cB_{\max}(\mathrm{ar}(\sigma)\cdot d)^2$ and by the choice of ε'. This contradicts the assumption that $\mathscr{D}$ is ε'-close to being induced $\mathscr{B}_i$-free and hence case (2) is not actually possible.

For case (3), let us assume that $\mathscr{B}_i$ contains at least one connected component isomorphic to $\mathscr{H}$ and at least one connected component that is not isomorphic to $\mathscr{H}$. Let $\mathscr{B}'_i$ be the database formed from $\mathscr{B}_i$ by removing every connected component isomorphic to $\mathscr{H}$. We will show that $\mathscr{D}_i$ is induced $\mathscr{B}'_i$-free and hence $\mathscr{D}'$ is also induced $\mathscr{B}'_i$-free. For a contradiction let us assume that $\mathscr{D}_i$ is not induced $\mathscr{B}'_i$-free. Since $\mathscr{D}_i$ is formed by removing or adding at most $\varepsilon' nd$ tuples from $\mathscr{D}$, $\mathscr{D}_i$ contains at least $n/c - \varepsilon' nd$ elements b such that $\mathscr{D}_i[b] \cong \mathscr{H}$. Using similar arguments to those used in case (2), if $\mathscr{D}_i$ contains at least $|B'_i|(\mathrm{ar}(\sigma)\cdot d)^2 + (|B_i| - |B'_i|)(\mathrm{ar}(\sigma)\cdot d)^2 = |B_i|(\mathrm{ar}(\sigma)\cdot d)^2$ elements $b \in D_i$ such that $\mathscr{D}_i[b] \cong \mathscr{H}$, then $\mathscr{D}_i$ will have an induced sub-database isomorphic to $\mathscr{B}_i$. However,

$$\frac{n}{c} - \varepsilon' nd = n\left(\frac{1}{c} - \varepsilon' d\right) \geq 2c|B_i|(\mathrm{ar}(\sigma)\cdot d)^2\left(\frac{1}{c} - \varepsilon' d\right) \geq |B_i|(\mathrm{ar}(\sigma)\cdot d)^2$$

since $n \geq 2cB_{\max}(\mathrm{ar}(\sigma)\cdot d)^2$ and $\varepsilon' \leq 1/2cd$. This contradicts that $\mathscr{D}_i$ is induced $\mathscr{B}_i$-free and so $\mathscr{D}_i$ must be induced $\mathscr{B}'_i$-free. Therefore $\mathscr{D}'$ is induced $\mathscr{B}'_i$-free and so we can insert

any number of disjoint copies of $\mathscr{H}$ into $\mathscr{D}'$ and the resulting database will still be induced $\mathscr{B}_i$-free.

Hence, if we insert $n - |D'|$ disjoint copies of $\mathscr{H}$ into $\mathscr{D}'$ then the resulting database $\mathscr{D}''$ will still be induced $\mathscr{B}_i$-free for all $\mathscr{B}_i \in B$ (and therefore $\mathscr{D}'' \in \mathbf{P}$). By the construction of $\mathscr{D}'$ and since for every $i \in [m]$, $\mathscr{D}_i$ is formed by adding or removing at most $\varepsilon' nd$ tuples from $\mathscr{D}$, $n - |D'| \leq m\varepsilon' nd$. Therefore

$$\operatorname{dist}(\mathscr{D}, \mathscr{D}'') \leq 2d(n - |D'|) \leq 2m\varepsilon' nd^2 = \frac{\varepsilon dn}{c} \leq \varepsilon dn.$$

This contradicts that $\mathscr{D}$ is ε-far from $\mathbf{P}$ and hence for at least one $i \in [m]$, $\mathscr{D}$ is ε'-far from being induced $\mathscr{B}_i$-free and so the tester will reject with probability at least $2/3$. □

This completes the proof of Theorem 3.3. The ε-tester for any property $\mathbf{P} \subseteq \mathbf{C}_d$ that is definable by a universal FO sentence ϕ consists of two steps. In the first step, the set of forbidden induced sub-databases B is constructed for $\mathbf{P}$ (as in Lemma 3.5). The ε-tester from the proof of Theorem 3.8 is then run on the input with the set B being the set of forbidden induced sub-databases of $\mathbf{P}$.

Note that we have also proved that any hereditary property which has a finite set of forbidden induced sub-databases is uniformly testable in constant time.

Chapter 4

Constant size databases that preserve the local structure of large databases

In this chapter we prove bounds on a result by Alon for two special cases. Alon [55, Proposition 19.10] proved that on bounded degree graphs, for any graph $\mathscr{G}$, radius r and $\varepsilon > 0$ there always exists a graph $\mathscr{H}$ whose size is independent of $|V(\mathscr{G})|$ and whose r-neighbourhood distribution vector satisfies $\|\operatorname{dv}_r(\mathscr{G}) - \operatorname{dv}_r(\mathscr{H})\|_1 \leq \varepsilon$.

Proposition 4.1 ([55])**.** *For every $\varepsilon > 0$ and constants d and r, there exists a positive integer $M(\varepsilon)$ such that for any d-degree bounded graph $\mathscr{G}$ there is a graph $\mathscr{H}$ on at most $M(\varepsilon)$ vertices such that $\|\operatorname{dv}_r(\mathscr{G}) - \operatorname{dv}_r(\mathscr{H})\|_1 \leq \varepsilon$.*

The proof of Proposition 4.1 is based on a compactness argument, however, the proof is only existential and does not provide an explicit bound on the size of $\mathscr{H}$. Finding such a bound was suggested by Alon as an open problem [47]. Currently, the only known bounds are for graphs with high-girth [32], where the girth of a graph $\mathscr{G}$ is the length of the shortest cycle in $\mathscr{G}$.

Proposition 4.1 easily extends to bounded degree databases and can be proved in a very similar way. We do not give the proof here, however, we will prove the existence of $\mathscr{H}$ and obtain explicit bounds on the size of $\mathscr{H}$ for certain classes of databases. We obtain explicit bounds for classes of graphs and databases of bounded degree whose histogram vectors form a semilinear set, or who are hyperfinite.

For the rest of this chapter, we fix a schema σ and a number $d \in \mathbb{N}$ with $d \geq 2$.

4.1 Databases with semilinear neighbourhood histograms

We will first obtain explicit bounds on the size of $\mathscr{H}$ for bounded degree databases that come from a class of databases whose neighbourhood histogram vectors form a semilinear set. Since graphs can be seen as special instances of databases, our bounds carry over to graphs also. Recall that $c(r) := |T_r^{\sigma,d}(1)|$ is the number of r-types with one centre and degree at most d, over schema σ.

Theorem 4.2. *Let $\varepsilon \in (0,1]$, let $r \in \mathbb{N}$ and let $\mathbf{C}$ be a class of σ-dbs of bounded degree d such that the set $\mathrm{h}_r(\mathbf{C})$ is semilinear. Let $\mathrm{h}_r(\mathbf{C}) = M_1 \cup M_2 \cup \cdots \cup M_m$ where $m \in \mathbb{N}$ and for each $i \in [m]$, $M_i = \{\vec{v}_0^i + a_1\vec{v}_1^i + \cdots + a_{k_i}\vec{v}_{k_i}^i \mid a_1, \ldots, a_{k_i} \in \mathbb{N}\}$ is a linear set where $\vec{v}_0^i, \ldots, \vec{v}_{k_i}^i \in \mathbb{N}^{\mathrm{c}(r)}$. Let $c := c(r)$, $k := \max_{i\in[m]} k_i + 1$ and $v := \max_{i\in[m]} \left(\max_{j\in[0,k_i]} \|\vec{v}_j^i\|_1\right)$. Then for any*

$$n_0 \geq kv\left(1 + \frac{3kvc}{\varepsilon}\right)$$

and $\mathscr{D} \in \mathbf{C}$ with $|D| > n_0 + kv$ there exists a σ-db $\mathscr{D}_0$ such that

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \varepsilon \text{ and } n_0 - kv \leq |D_0| \leq n_0 + kv.$$

Furthermore, $\mathscr{D}_0 \in \mathbf{C}$.

Proof. Let $n_0 \geq kv(1 + 3kvc/\varepsilon)$ and let $\mathscr{D} \in \mathbf{C}$ with $|D| = n > n_0 + kv$. Then there exists some $i \in [m]$ and $a_1^{\mathscr{D}}, \ldots, a_{k_i}^{\mathscr{D}} \in \mathbb{N}$ such that $\mathrm{h}_r(\mathscr{D}) = \vec{v}_0^i + a_1^{\mathscr{D}}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}}\vec{v}_{k_i}^i$ (note that $n = \|\vec{v}_0^i\|_1 + \sum_{j\in[k_i]} a_j^{\mathscr{D}}\|\vec{v}_j^i\|_1$). Let $\mathscr{D}_0$ be the σ-db with r-histogram $\vec{v}_0^i + a_1^{\mathscr{D}_0}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}_0}\vec{v}_{k_i}^i \in M_i$ where $a_j^{\mathscr{D}_0}$ is the nearest integer to $a_j^{\mathscr{D}} n_0/n$ (if $a_j^{\mathscr{D}} n_0/n$ is precisely between two integers, then just choose the smallest), and hence

$$\frac{a_j^{\mathscr{D}} n_0}{n} - \frac{1}{2} \leq a_j^{\mathscr{D}_0} \leq \frac{a_j^{\mathscr{D}} n_0}{n} + \frac{1}{2}.$$

Note that since $\vec{v}_0^i + a_1^{\mathscr{D}_0}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}_0}\vec{v}_{k_i}^i \in \mathrm{h}_r(\mathbf{C})$, $\mathscr{D}_0$ exists and $\mathscr{D}_0 \in \mathbf{C}$. We need to show that $n_0 - kv \leq |D_0| \leq n_0 + kv$ and $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \varepsilon$.

Claim 4.3. $|D_0| \geq n_0 - kv$.

Proof: By the choice of $a_j^{\mathscr{D}_0}$ for $j \in [k_i]$,

$$|D_0| = \|\vec{v}_0^i\|_1 + \sum_{j\in[k_i]} a_j^{\mathscr{D}_0}\|\vec{v}_j^i\|_1$$

$$
\begin{aligned}
&\geq \|\vec{v}_0^i\|_1 + \sum_{j\in[k_i]} \Big(\frac{a_j^{\mathscr{D}} n_0}{n} - \frac{1}{2}\Big)\|\vec{v}_j^i\|_1 \\
&= \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{j\in[k_i]} \|\vec{v}_j^i\|_1 + \frac{n_0}{n}\sum_{j\in[k_i]} a_j^{\mathscr{D}}\|\vec{v}_j^i\|_1 \\
&= \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{j\in[k_i]} \|\vec{v}_j^i\|_1 + n_0 - \frac{n_0\|\vec{v}_0^i\|_1}{n} \\
&\geq \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{j\in[k_i]} \|\vec{v}_j^i\|_1 + n_0 - \|\vec{v}_0^i\|_1 \\
&\geq -kv + n_0,
\end{aligned}
$$

as $\sum_{j\in[k_i]} a_j^{\mathscr{D}}\|\vec{v}_j^i\|_1 = n - \|\vec{v}_0^i\|_1$ and $n > n_0$. ■

Claim 4.4. $|D_0| \leq n_0 + kv$.

Proof: By the choice of $a_j^{\mathscr{D}_0}$ for $j \in [k_i]$,

$$
\begin{aligned}
|D_0| &= \|\vec{v}_0^i\|_1 + \sum_{j\in[k_i]} a_j^{\mathscr{D}_0}\|\vec{v}_j^i\|_1 \\
&\leq \|\vec{v}_0^i\|_1 + \sum_{j\in[k_i]} \Big(\frac{a_j^{\mathscr{D}} n_0}{n} + \frac{1}{2}\Big)\|\vec{v}_j^i\|_1 \\
&= \|\vec{v}_0^i\|_1 + \frac{1}{2}\sum_{j\in[k_i]} \|\vec{v}_j^i\|_1 + n_0\Big(1 - \frac{\|\vec{v}_0^i\|_1}{n}\Big) \\
&\leq \sum_{0\leq j\leq k_i} \|\vec{v}_j^i\|_1 + n_0 \\
&\leq kv + n_0,
\end{aligned}
$$

as $\sum_{j\in[k_i]} a_j^{\mathscr{D}}\|\vec{v}_j^i\|_1 = n - \|\vec{v}_0^i\|_1$. ■

Claim 4.5. $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \varepsilon$.

Proof: By definition, $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 = \sum_{j\in[c]} |\operatorname{dv}_r(\mathscr{D})[j] - \operatorname{dv}_r(\mathscr{D}_0)[j]|$. First recall that $0 < n_0 - kv \leq |D_0| \leq n_0 + kv < n$ (by the choice of n_0 and by Claims 4.3 and 4.4) and note that for every $\ell \in [k_i]$, $a_\ell^{\mathscr{D}} \leq n$ (since $\|\vec{v}_\ell^i\|_1 \neq 0$). For every $j \in [c]$, by the choice of $a_\ell^{\mathscr{D}_0}$ for $\ell \in [k_i]$,

$$
\begin{aligned}
&\operatorname{dv}_r(\mathscr{D})[j] - \operatorname{dv}_r(\mathscr{D}_0)[j] \\
&= \vec{v}_0^i[j]\Big(\frac{1}{n} - \frac{1}{|D_0|}\Big) + \sum_{\ell\in[k_i]} \vec{v}_\ell^i[j]\Big(\frac{a_\ell^{\mathscr{D}}}{n} - \frac{a_\ell^{\mathscr{D}_0}}{|D_0|}\Big)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}}_\ell}{n} - \frac{a^{\mathscr{D}}_\ell n_0}{n|D_0|} + \frac{1}{2|D_0|}\Big)\\
&= \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}}_\ell}{n}\Big(\frac{|D_0|-n_0}{|D_0|}\Big) + \frac{1}{2|D_0|}\Big)\\
&\leq \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{n}{n}\Big(\frac{kv+n_0-n_0}{n_0-kv}\Big) + \frac{1}{2(n_0-kv)}\Big)\\
&= \Big(\frac{2kv+1}{2(n_0-kv)}\Big)\sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\\
&\leq \frac{kv(2kv+1)}{n_0-kv}.
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
&\mathrm{dv}_r(\mathscr{D})[j] - \mathrm{dv}_r(\mathscr{D}_0)[j]\\
&\geq -\frac{\vec{v}^i_0[j]}{|D_0|} + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}}_\ell}{n}\Big(\frac{|D_0|-n_0}{|D_0|}\Big) - \frac{1}{2|D_0|}\Big)\\
&\geq -\frac{\vec{v}^i_0[j]}{|D_0|} + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}}_\ell}{n}\Big(\frac{-kv+n_0-n_0}{|D_0|}\Big) - \frac{1}{2|D_0|}\Big)\\
&= -\frac{\vec{v}^i_0[j]}{|D_0|} - \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}}_\ell kv}{n|D_0|} + \frac{1}{2|D_0|}\Big)\\
&\geq -\frac{\vec{v}^i_0[j]}{n_0-kv} - \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{nkv}{n(n_0-kv)} + \frac{1}{2(n_0-kv)}\Big)\\
&= -\frac{\vec{v}^i_0[j]}{n_0-kv} - \Big(\frac{2kv+1}{2(n_0-kv)}\Big)\sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\\
&\geq -\frac{kv(2kv+1)}{n_0-kv}.
\end{aligned}
$$

Hence,

$$|\mathrm{dv}_r(\mathscr{D})[j] - \mathrm{dv}_r(\mathscr{D}_0)[j]| \leq \frac{kv(2kv+1)}{n_0-kv} \leq \frac{3(kv)^2}{n_0-kv} \leq \frac{\varepsilon}{c}$$

by the choice of n_0. Therefore,

$$\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}_0)\|_1 = \sum_{j\in[c]} |\mathrm{dv}_r(\mathscr{D})[j] - \mathrm{dv}_r(\mathscr{D}_0)[j]| \leq \varepsilon$$

as required. ■

The above three claims complete the proof. □

We will make use of Theorem 4.2 in Chapter 5 to construct constant time property testers (in the new model introduced in Chapter 5) for any property **P** which is hyperfinite and has a semilinear set of neighbourhood histograms. In Chapter 5 we prove that for such properties, for any database $\mathscr{D}$ which is far from the property (in the new model), there does not exist a small constant size database in the property with a similar distribution vector to $\mathscr{D}$ (note that this result requires the property to be hyperfinite). Combining this result with Theorem 4.2, we can construct a constant time tester for any hyperfinite property with a semilinear set of neighbourhood histograms as follows: Given an input database, we make a good estimate of its neighbourhood distribution vector (which can be done with high probability by Lemma 2.19) and then the tester accepts if this estimate is close to the distribution vector of some small constant size database in the property, otherwise it rejects. We point out here that to be able to distinguish between inputs in the property from those that are far away from the property, the small constant size database $\mathscr{D}_0$ constructed in Theorem 4.2 must belong to the same class as the large database $\mathscr{D}$.

4.2 Hyperfinite databases

In Lemma 2.38 we proved that if **C** is the class of σ-dbs of bounded degree d, whose connected components all have at most k elements (for some $k \in \mathbb{N}$), then $\mathrm{h}_r(\mathbf{C})$ is linear for any $r \in \mathbb{N}$. Since any hyperfinite database is close to some database in **C** (assuming k is chosen carefully), we can combine Lemma 2.38 and Theorem 4.2 to obtain explicit bounds on the size of $\mathscr{H}$ for hyperfinite databases (Theorem 4.6). We note that our bounds carry over to graphs also. Whilst we do not directly use Theorem 4.6 in our testers in the following chapters, we believe this result is of independent interest.

Recall that $\Psi(k,\sigma)$ is a maximal set of non-isomorphic connected σ-dbs of size at most k.

Theorem 4.6. *Let $\varepsilon \in (0,1]$, let $r \in \mathbb{N}$ and let $\mathscr{D}$ be a σ-db of bounded degree d that is ρ-hyperfinite for some function ρ. Let*

$$\varepsilon_0 := \frac{\varepsilon}{4(\mathrm{ar}(\sigma)\cdot d)^{r+1}},$$

let $v := \rho(\varepsilon_0)$, let $k := |\Psi(v,\sigma)| + 1$ and let $c := c(r)$. Then for any

$$n_0 \geq kv\Big(1 + \frac{6kvc}{\varepsilon}\Big)$$

if $|D| > n_0 + kv$, then there exists a σ-db $\mathscr{D}_0$ such that

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \varepsilon \text{ and } n_0 - kv \leq |D_0| \leq n_0 + kv.$$

Proof. Let $n_0 \geq kv(1+6kvc/\varepsilon)$ and let $n := |D| > n_0 + kv$. Since $\mathscr{D}$ is ρ-hyperfinite and by the choice of v, we can remove at most $\varepsilon_0 n$ tuples from $\mathscr{D}$ to obtain a σ-db $\mathscr{D}'$ (on n elements) whose connected components are all of size at most v. Let $\bar{a} = (a_1, \ldots, a_\ell)$ be a tuple that was removed from $\mathscr{D}$ to form $\mathscr{D}'$. For any element $b \in D$, if $\bar{a}$ is in the r-neighbourhood of b in $\mathscr{D}$, then the r-type of b in $\mathscr{D}'$ could be different from the r-type of b in $\mathscr{D}$. The tuple $\bar{a}$ is in the r-neighbourhood of b in $\mathscr{D}$ if $a_1, \ldots, a_\ell \in N_r^{\mathscr{D}}(b)$, or alternatively if $b \in N_r^{\mathscr{D}}(a_1) \cap \cdots \cap N_r^{\mathscr{D}}(a_\ell)$. By Lemma 2.16, $|N_r(a_1) \cap N_r(a_2) \cap \cdots \cap N_r(a_\ell)| \leq |N_r(a_1)| \leq (\operatorname{ar}(\sigma) \cdot d)^{r+1}$, and hence every tuple that was removed from $\mathscr{D}$ to form $\mathscr{D}'$ could have changed the r-type of at most $(\operatorname{ar}(\sigma) \cdot d)^{r+1}$ elements. Therefore

$$\|\operatorname{h}_r(\mathscr{D}) - \operatorname{h}_r(\mathscr{D}')\|_1 \leq 2\varepsilon_0 n (\operatorname{ar}(\sigma) \cdot d)^{r+1}$$

and

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq 2\varepsilon_0 (\operatorname{ar}(\sigma) \cdot d)^{r+1}.$$

Let $\mathbf{C}$ be the class of all σ-dbs of bounded degree d whose connected components are all of size at most v. Let $\Psi := \Psi(v, \sigma) = \{\mathscr{D}_1, \ldots \mathscr{D}_{|\Psi|}\}$. By Lemma 2.38, $\operatorname{h}_r(\mathbf{C})$ is a linear set and can be written as

$$\operatorname{h}_r(\mathbf{C}) = \{a_1 \operatorname{h}_r(\mathscr{D}_1) + a_2 \operatorname{h}_r(\mathscr{D}_2) + \cdots + a_{|\Psi|} \operatorname{h}_r(\mathscr{D}_{|\Psi|}) \mid a_1, \ldots a_{|\Psi|} \in \mathbb{N}\}.$$

Since $\mathscr{D}' \in \mathbf{C}$ and by the choice of k, v and n_0, by Theorem 4.2 there exists a σ-db $\mathscr{D}_0$ such that

$$\|\operatorname{dv}_r(\mathscr{D}') - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \frac{\varepsilon}{2} \text{ and } n_0 - kv \leq |D_0| \leq n_0 + kv.$$

Finally,

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq \frac{\varepsilon}{2} + 2\varepsilon_0 (\operatorname{ar}(\sigma) \cdot d)^{r+1} = \varepsilon$$

by the choice of ε_0. □

In Theorem 4.2, the constant size database $\mathscr{D}_0$ constructed also belongs to the same class of databases as the large database $\mathscr{D}$. In section 4.1, we discussed how this fact will be essential when we use Theorem 4.2 in Chapter 5 to construct our testers. However, for ρ-hyperfinite databases (for some function ρ), it is not clear whether the constant size database $\mathscr{D}_0$ constructed in Theorem 4.6 is ρ-hyperfinite too. We will discuss this now.

Let $\varepsilon \in (0,1]$, let $r \in \mathbb{N}$ and let $\mathscr{D}$ be a d-degree bounded σ-db that is ρ-hyperfinite (for some function ρ). Let ε_0, v, k, c and n_0 be as in Theorem 4.6. Let us assume that $|D| := n > n_0 + kv$ and let $\mathscr{D}_0$ be the σ-db found in Theorem 4.6. To construct $\mathscr{D}_0$, first we obtained the σ-db $\mathscr{D}'$ which is formed from $\mathscr{D}$ by removing at most $\varepsilon_0 n$ tuples such that in $\mathscr{D}'$ every connected component has size at most v. Clearly $\mathscr{D}'$ is ρ-hyperfinite. We then construct $\mathscr{D}_0$ from $\mathscr{D}'$ by applying Theorem 4.2. Let $\mathbf{C}$ be the class of all d-degree bounded σ-dbs whose connected components are all of size at most v. Let $\Psi := \Psi(v,\sigma) = \{\mathscr{D}_1, \ldots \mathscr{D}_{|\Psi|}\}$. Then for $i \in [|\Psi|]$, let $a_i^{\mathscr{D}'}$ be the number of connected components in $\mathscr{D}'$ isomorphic to $\mathscr{D}_i$. Then

$$\mathrm{h}_r(\mathscr{D}') = a_1^{\mathscr{D}'}\mathrm{h}_r(\mathscr{D}_1) + a_2^{\mathscr{D}'}\mathrm{h}_r(\mathscr{D}_2) + \cdots + a_{|\Psi|}^{\mathscr{D}'}\mathrm{h}_r(\mathscr{D}_{|\Psi|}).$$

By the proof of Theorem 4.2, $\mathscr{D}_0$ is the σ-db with r-histogram vector

$$\mathrm{h}_r(\mathscr{D}_0) = a_1^{\mathscr{D}_0}\mathrm{h}_r(\mathscr{D}_1) + a_2^{\mathscr{D}_0}\mathrm{h}_r(\mathscr{D}_2) + \cdots + a_{|\Psi|}^{\mathscr{D}_0}\mathrm{h}_r(\mathscr{D}_{|\Psi|})$$

where for $i \in [|\Psi|]$, $a_i^{\mathscr{D}_0}$ is the closest integer to $a_i^{\mathscr{D}'} n_0/n$. In particular $\mathscr{D}_0$ is the σ-db with exactly $a_i^{\mathscr{D}_0}$ connected components isomorphic to $\mathscr{D}_i$ for $i \in [|\Psi|]$. To show that $\mathscr{D}_0$ is ρ-hyperfinite we need to show that for every $\varepsilon' > 0$, $\mathscr{D}_0$ is $(\varepsilon', \rho(\varepsilon'))$-hyperfinite. Let $\varepsilon' > 0$. If $\rho(\varepsilon') \geq v$ then $\mathscr{D}_0$ is $(\varepsilon', \rho(\varepsilon'))$-hyperfinite. So let us assume that $\rho(\varepsilon') < v$. In the worse case scenario we need to remove $\varepsilon' n$ tuples from $\mathscr{D}'$ to obtain a σ-db whose connected components are all of size at most $\rho(\varepsilon')$. Let us assume this is the case. For each $\mathscr{D}_i \in \Psi$, let b_i be the minimum number of tuples that are needed to be removed from $\mathscr{D}_i$ to form a σ-db whose connected components are all of size at most $\rho(\varepsilon')$. Hence

$$\sum_{i \in [|\Psi|]} a_i^{\mathscr{D}'} b_i = \varepsilon' n.$$

Furthermore, to form a σ-db from $\mathscr{D}_0$ whose connected components are all of size at most $\rho(\varepsilon')$ we need to remove

$$\sum_{i \in [|\Psi|]} a_i^{\mathscr{D}_0} b_i$$

many tuples. However in the worse case scenario, $a_i^{\mathscr{D}_0} = a_i^{\mathscr{D}'} n_0/n + 1/2$ (since $a_i^{\mathscr{D}_0} \leq a_i^{\mathscr{D}'} n_0/n + 1/2$) for every $i \in [|\Psi|]$ and $n_0 - kv = |D_0|$ (since $n_0 - kv \leq |D_0|$). If this is the case then

$$\sum_{i \in [|\Psi|]} a_i^{\mathscr{D}_0} b_i = \sum_{i \in [|\Psi|]} \Big(\frac{a_i^{\mathscr{D}'} n_0}{n} + \frac{1}{2}\Big) b_i = \varepsilon' n_0 + \frac{1}{2} \sum_{i \in [|\Psi|]} b_i = \varepsilon' |D_0| + kv\varepsilon' + \frac{1}{2} \sum_{i \in [|\Psi|]} b_i > \varepsilon' |D_0|.$$

Therefore we cannot be certain whether $\mathscr{D}_0$ is ρ-hyperfinite.

Chapter 5

A new property testing model

Much research has focussed on the query complexity of property testing algorithms, but in view of applications, the running time of the algorithm is equally relevant. In [4], in the bounded degree relational database model (the BDRD model), it was shown that on databases of bounded degree and bounded tree-width, every property that is expressible in monadic second-order logic with counting (CMSO) is testable with constant query complexity and polylogarithmic running time (Theorem 2.35). It remains open whether this can be improved to constant running time.

In this chapter, we introduce a new model, which is based on the BDRD model, but the distance measure allows both tuple modifications and element modifications (we call our model the $\text{BDRD}_{+/-}$ model for short). We show that every property that is testable in the BDRD model is testable in the $\text{BDRD}_{+/-}$ model with the same query complexity and running time, but the converse is not true (Lemmas 5.3 and 5.5). Our main theorem shows that on databases of bounded degree and bounded tree-width, every property that is expressible in CMSO is testable with constant query complexity and constant running time in the $\text{BDRD}_{+/-}$ model (Theorem 5.15). Our proof methods include the semilinearity of the neighbourhood histograms of databases having the property and the result by Alon [55, Proposition 19.10] that states that for every bounded degree graph $\mathscr{G}$ there exists a constant size graph $\mathscr{H}$ that has a similar neighbourhood distribution to $\mathscr{G}$.

We actually prove two more general results than Theorem 5.15. First, we prove that any hyperfinite property whose histogram vectors form a semilinear set are uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model (Theorem 5.14). We also prove that being hyperfinite and having a set of histogram vectors that are close to being semilinear is enough for constant time uniform testability in the $\text{BDRD}_{+/-}$ model (Theorem 5.19).

The authors in [17] show that monotone hyperfinite properties are testable with constant query complexity and constant running time in the bounded degree graph model. It can be

derived from this result that hyperfinite hereditary properties are testable with constant query complexity and constant running time in the BDRD model (and hence also in the $\text{BDRD}_{+/-}$ model). Using our methods, we give alternative proofs that hyperfinite monotone properties are testable with constant query complexity and constant running time in the BDRD and $\text{BDRD}_{+/-}$ models, and that hyperfinite hereditary properties are testable with constant query complexity and constant running time in the $\text{BDRD}_{+/-}$ model.

In Section 5.1 we introduce our property testing model for relational databases of bounded degree and we compare it to the BDRD model. In Section 5.2 we define a notion of locality in both the BDRD and $\text{BDRD}_{+/-}$ models and prove that hyperfinite properties are local in the $\text{BDRD}_{+/-}$ model. In Section 5.3 we prove our main theorems. In Section 5.4 we prove a more general theorem of that proved in Section 5.3. In Section 5.5 we give an alternative proof of the constant time uniform testability of hyperfinite hereditary properties in the $\text{BDRD}_{+/-}$ model. Finally, in Section 5.6 we give an alternative proof of the constant time uniform testability of monotone hereditary properties in the BDRD model.

Proviso. *For the rest of the chapter, we fix a schema σ and numbers $d,t \in \mathbb{N}$ with $d \geq 2$. All databases are σ-dbs and have degree at most d, unless stated otherwise. We use $\mathbf{C}_d$ to denote the class of all σ-dbs with degree at most d, $\mathbf{C}_d^t$ to denote the class of all σ-dbs with degree at most d and tree-width at most t and finally we use $\mathbf{C}$ to denote a class of σ-dbs with degree at most d (which is closed under isomorphism).*

5.1 The model

We will now introduce our property testing model for bounded degree relational databases, which is an extension of the BDRD model introduced in Section 2.6. The notions of oracle queries, properties, ε-tester, query complexity and uniform testability remain the same but we have an alternative definition of distance and ε-closeness. In our model, which we shall call the $\text{BDRD}_{+/-}$ *model* for short, we can add and remove elements as well as tuples and can therefore compare databases that are on a different number of elements.

Definition 5.1 (Distance and ε-closeness)**.** *Let $\mathscr{D},\mathscr{D}' \in \mathbf{C}_d$ and $\varepsilon \in [0,1]$. The distance between $\mathscr{D}$ and $\mathscr{D}'$ (denoted by $\text{dist}_{+/-}(\mathscr{D},\mathscr{D}')$) is the minimum number of modifications we need to make to $\mathscr{D}$ and $\mathscr{D}'$ to make them isomorphic where a modification is either (1) inserting a new element, (2) deleting an element (and as a result deleting any tuple that contains that element), (3) inserting a tuple, or (4) deleting a tuple. We then say $\mathscr{D}$ and $\mathscr{D}'$ are ε-close if $\text{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \varepsilon d \min\{|D|,|D'|\}$ and are ε-far otherwise.*

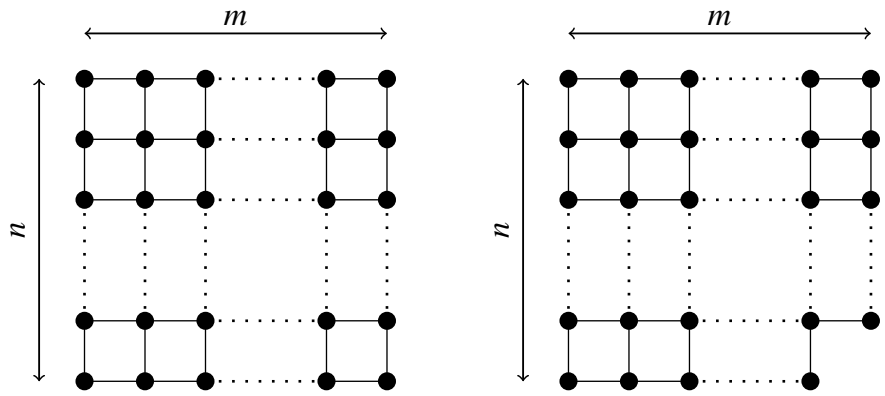

Fig. 5.1 The graphs $\mathscr{G}_{n,m}$ and $\mathscr{H}_{n,m}$ (respectively) of Example 5.2.

The following example illustrates the difference between the distance measure of the BDRD and the distance measure of the $\text{BDRD}_{+/-}$ model.

Example 5.2. Let $\mathbf{P} = \{\mathscr{G}_{n,m} \mid n,m \in \mathbb{N}_{>1}\}$ where $\mathscr{G}_{n,m}$ is an n by m grid graph as shown in Figure 5.1. Let us consider the graph $\mathscr{H}_{n,m}$ for some $n,m \in \mathbb{N}_{>1}$ which is formed from $\mathscr{G}_{n,m}$ by removing a corner vertex. In the $\text{BDRD}_{+/-}$ model the distance between $\mathscr{H}_{n,m}$ and $\mathscr{G}_{n,m}$ is 1 (we remove a corner vertex from $\mathscr{G}_{n,m}$ to get $\mathscr{H}_{n,m}$) and therefore $\mathscr{H}_{n,m}$ is at distance 1 from $\mathbf{P}$ in the $\text{BDRD}_{+/-}$ model. In the BDRD model if two graphs are on a different number of vertices then the distance between them is infinity. Therefore if $nm-1$ is a prime number then $\mathscr{H}_{n,m}$ is at distance infinity from $\mathbf{P}$ in the BDRD model.

We now show that if a property is uniformly testable in the BDRD model then it is also uniformly testable in the $\text{BDRD}_{+/-}$ model but the converse is not true. This allows for more uniformly testable properties in the $\text{BDRD}_{+/-}$ model.

Lemma 5.3. *Let $\mathbf{P} \subseteq \mathbf{C}$ be a property on $\mathbf{C}$. If $\mathbf{P}$ is uniformly testable on $\mathbf{C}$ in time $f(n)$ in the* BDRD *model then $\mathbf{P}$ is also uniformly testable on $\mathbf{C}$ in time $f(n)$ in the* $\text{BDRD}_{+/-}$ *model.*

Proof. Let $\varepsilon \in (0,1]$. Let π be an ε-tester, that runs in time $f(n)$, for $\mathbf{P}$ on $\mathbf{C}$ in the BDRD model. We claim that π is also an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ in the $\text{BDRD}_{+/-}$ model. Let $\mathscr{D} \in \mathbf{C}$ be the input σ-db. If $\mathscr{D} \in \mathbf{P}$ then π will accept with probability at least $2/3$. If $\mathscr{D}$ is ε-far from $\mathbf{P}$ in the $\text{BDRD}_{+/-}$ model then it must also be ε-far from $\mathbf{P}$ in the BDRD model and therefore π will reject with probability at least $2/3$. Hence π is an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ in the $\text{BDRD}_{+/-}$ model. □

Theorem 5.4 ([42])**.** *In the bounded degree graph model, bipartiteness cannot be tested with query complexity $o(\sqrt{n})$, where n is the number of vertices of the input graph.*

Lemma 5.5. *There exists a class* $\mathbf{C}$ *of* σ*-dbs and a property* $\mathbf{P} \subseteq \mathbf{C}$ *such that* $\mathbf{P}$ *is trivially uniformly testable on* $\mathbf{C}$ *in the* $\text{BDRD}_{+/-}$ *model but is not testable on* $\mathbf{C}$ *in the* BDRD *model.*

Proof. Let $\mathbf{C}$ be the class of all graphs with degree at most d. Let $\mathbf{P} = \mathbf{P_1} \cup \mathbf{P_2} \subseteq \mathbf{C}$ be the property where $\mathbf{P_1}$ contains all bipartite graphs in $\mathbf{C}$ and $\mathbf{P_2}$ contains all graphs in $\mathbf{C}$ that have an odd number of vertices. In the $\text{BDRD}_{+/-}$ model every $\mathscr{G} \in \mathbf{C}$ is ε-close to $\mathbf{P}$ if $|V(\mathscr{G})| \geq 1/(\varepsilon d)$ and hence $\mathbf{P}$ is trivially testable on $\mathbf{C}$ in the $\text{BDRD}_{+/-}$ model (the tester accepts if $|V(\mathscr{G})| \geq 1/(\varepsilon d)$ and does a full check of the input otherwise). In the BDRD model, if the input graph has an even number of vertices then it is far from $\mathbf{P_2}$ and so we have to test for $\mathbf{P_1}$. By Theorem 5.4, bipartiteness is not testable (with constant query complexity) in the BDRD model. In particular, in the proof of Theorem 5.4, Goldreich and Ron show that for any even n there exists two families, $\mathscr{G}_1 \subseteq \mathbf{C}$ and $\mathscr{G}_2 \subseteq \mathbf{C}$, of n-vertex graphs such that every graph in $\mathscr{G}_1$ is bipartite and almost all graphs in $\mathscr{G}_2$ are far from being bipartite but any algorithm that performs $o(\sqrt{n})$ queries cannot distinguish between a graph chosen randomly from $\mathscr{G}_1$ and a graph chosen randomly from $\mathscr{G}_2$. Therefore $\mathbf{P}$ is not testable on $\mathbf{C}$ in the BDRD model. □

Note that the underlying general principle of the above proof can be applied to obtain further examples of properties that are testable in the $\text{BDRD}_{+/-}$ model but not testable in the BDRD model.

5.2 Locality of properties

It is known that every hyperfinite property is 'local' in the BDRD model (Theorem 5.7), where a property is 'local' if a σ-db $\mathscr{D}$ has a similar r-histogram to some σ-db (with the same domain size) that has the property, then $\mathscr{D}$ must be ε-close to the property [57, 4]. This is summarised in Definition 5.6 and Theorem 5.7 below. We define an equivalent definition of locality in the $\text{BDRD}_{+/-}$ model (Definition 5.8). We prove that any property that is local in the BDRD model is also local in the $\text{BDRD}_{+/-}$ model (Lemma 5.9) and hence every hyperfinite property is local in the $\text{BDRD}_{+/-}$ model (Theorem 5.10). Theorem 5.10 is essential for the proof of Theorem 5.14.

Definition 5.6 (Locality in the BDRD model)**.** *Let* $\varepsilon \in (0,1]$. *A property* $\mathbf{P} \subseteq \mathbf{C}$ *is* ε-local *on* $\mathbf{C}$ *in the* BDRD *model if there exists* $\lambda := \lambda_{5.6}(\varepsilon) \in (0,1]$, $r := r_{5.6}(\varepsilon) \in \mathbb{N}$ *and* $N := N_{5.6}(\varepsilon) \in \mathbb{N}$ [1] *such that for each* $\mathscr{D} \in \mathbf{P}$ *and* $\mathscr{D}' \in \mathbf{C}$ *with the same number* $n \geq N$ *of elements, if* $\|h_r(\mathscr{D}) - h_r(\mathscr{D}')\|_1 \leq \lambda n$, *then* $\mathscr{D}'$ *is* ε*-close to* $\mathbf{P}$ *in the* BDRD *model.*

[1] Note that we use the definition number as a subscript so we can clearly refer to these parameters later.

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

Theorem 5.7 ([57, 4])**.** *Let* **C** *be closed under removing tuples. If a property* $\mathbf{P} \subseteq \mathbf{C}$ *is hyperfinite on* **C***, then* **P** *is local on* **C** *in the* BDRD *model.*

We now define the notion of locality in the $\mathrm{BDRD}_{+/-}$ model.

Definition 5.8 (Locality in the $\mathrm{BDRD}_{+/-}$ model)**.** *Let $\varepsilon \in (0,1]$. A property* $\mathbf{P} \subseteq \mathbf{C}$ *is ε-*local *on* **C** *in the* $\mathrm{BDRD}_{+/-}$ *model if there exists $\lambda := \lambda_{5.8}(\varepsilon) \in (0,1]$, $r := r_{5.8}(\varepsilon) \in \mathbb{N}$ and $N := N_{5.8}(\varepsilon) \in \mathbb{N}$ such that for each $\mathscr{D} \in \mathbf{P}$ and $\mathscr{D}' \in \mathbf{C}$, on $|D| \geq N$ and $|D'| \geq N$ elements respectively, if $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq \lambda \min\{|D|,|D'|\}$, then $\mathscr{D}'$ is ε-close to* **P** *in the* $\mathrm{BDRD}_{+/-}$ *model.*

We call the parameters r and λ the locality radius *and* disc proximity *of* **P** *for ε, respectively. A property is* local *in the* $\mathrm{BDRD}_{+/-}$ *model if it is ε-local in the* $\mathrm{BDRD}_{+/-}$ *model for every $\varepsilon \in (0,1]$.*

Lemma 5.9. *Let* **C** *be closed under removing and inserting elements. If a property* $\mathbf{P} \subseteq \mathbf{C}$ *is local on* **C** *in the* BDRD *model, then* **P** *is local on* **C** *in the* $\mathrm{BDRD}_{+/-}$ *model.*

Proof. Let $\varepsilon \in (0,1]$. Let $r_{5.6}(\varepsilon/4)$, $\lambda_{5.6}(\varepsilon/4)$ and $N_{5.6}(\varepsilon/4)$ be as in Definition 5.6 for **P** and $\varepsilon/4$. Let $r := r_{5.6}(\varepsilon/4)$, let $N := N_{5.6}(\varepsilon/4)$ and let

$$\lambda := \frac{\varepsilon \lambda_{5.6}(\varepsilon/4)}{1 + 2(\mathrm{ar}(\sigma) \cdot d)^{r+1}}.$$

We will prove that **P** is ε-local on **C** in the $\mathrm{BDRD}_{+/-}$ model with $r_{5.8}(\varepsilon) = r$, $\lambda_{5.8}(\varepsilon) = \lambda$ and $N_{5.8}(\varepsilon) = N$.

Let $\mathscr{D} \in \mathbf{P}$ and $\mathscr{D}' \in \mathbf{C}$ where $|D| \geq N$ and $|D'| \geq N$. Let us assume that $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq \lambda \min\{|D|,|D'|\}$ and **P** is local on **C** in the BDRD model. We will show that $\mathscr{D}'$ is ε-close to **P**.

If $|D| = |D'|$ then since $\lambda \leq \lambda_{5.6}(\varepsilon/4)$, $r = r_{5.6}(\varepsilon/4)$ and $N = N_{5.6}(\varepsilon/4)$, $\mathscr{D}'$ is $\varepsilon/4$-close to **P** and hence $\mathscr{D}'$ is also ε-close to **P**. So let us assume that $|D| \neq |D'|$. Let $\mathscr{D}_1$ be the σ-db on $|D|$ elements formed from $\mathscr{D}'$ by either removing $|D'| - |D|$ elements if $|D| < |D'|$ or adding $|D| - |D'|$ new elements if $|D'| < |D|$ (note that $\mathscr{D}_1 \in \mathbf{C}$). Note that as $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq \lambda \min\{|D|,|D'|\}$ and by definition $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 = \sum_{i=1}^{\mathrm{c}(r)} |\mathrm{h}_r(\mathscr{D})[i] - \mathrm{h}_r(\mathscr{D}')[i]|$ we have $\big||D| - |D'|\big| \leq \lambda \min\{|D|,|D'|\}$. When an element is inserted no other elements r-type will change, however, when an element a is removed, the r-type of any element in $N_r(a)$ will

change. Since $|N_r(a)| \leq (\mathrm{ar}(\sigma)\cdot d)^{r+1}$ (by Lemma 2.16) and $\big||D|-|D'|\big| \leq \lambda \min\{|D|,|D'|\}$, we have $\|\mathrm{h}_r(\mathscr{D}')-\mathrm{h}_r(\mathscr{D}_1)\|_1 \leq 2\lambda \min\{|D|,|D'|\}(\mathrm{ar}(\sigma)\cdot d)^{r+1}$. Therefore

$$\|\mathrm{h}_r(\mathscr{D})-\mathrm{h}_r(\mathscr{D}_1)\|_1 \leq \lambda \min\{|D|,|D'|\}(1+2(\mathrm{ar}(\sigma)\cdot d)^{r+1}) \leq \lambda_{5.6}(\varepsilon/4)|D|$$

by the choice of λ. Since $\mathbf{P}$ is local on $\mathbf{C}$ in the BDRD model, $|D|=|D_1|$ and $\mathscr{D}\in\mathbf{P}$, $\mathscr{D}_1$ is $\varepsilon/4$-close to $\mathbf{P}$ in the BDRD model. Hence there exists a σ-db $\mathscr{D}_2\in\mathbf{P}$ such that $|D_2|=|D|$ and $\mathrm{dist}(\mathscr{D}_1,\mathscr{D}_2)\leq \varepsilon d|D|/4$. By the definition of the two distance measures dist and $\mathrm{dist}_{+/-}$, we have $\mathrm{dist}_{+/-}(\mathscr{D}_1,\mathscr{D}_2)\leq \mathrm{dist}(\mathscr{D}_1,\mathscr{D}_2)\leq \varepsilon d|D|/4$ and by the construction of $\mathscr{D}_1$ we have $\mathrm{dist}_{+/-}(\mathscr{D}',\mathscr{D}_1)\leq \lambda\min\{|D|,|D'|\}=\lambda\min\{|D'|,|D_2|\}$. Therefore

$$\mathrm{dist}_{+/-}(\mathscr{D}',\mathscr{D}_2)\leq \frac{\varepsilon d|D|}{4}+\lambda\min\{|D'|,|D_2|\}\leq \varepsilon d\min\{|D'|,|D_2|\},$$

since $\lambda\leq \varepsilon d/2$ and since

$$|D|\leq (1+\lambda)\min\{|D|,|D'|\}\leq 2\min\{|D|,|D'|\}=2\min\{|D'|,|D_2|\}$$

(if $|D|<|D'|$ then clearly this holds, otherwise since $\big||D|-|D'|\big|\leq\lambda\min\{|D|,|D'|\}$, $|D|\leq |D'|+\lambda\min\{|D|,|D'|\}=(1+\lambda)\min\{|D|,|D'|\}$). Hence in the $\mathrm{BDRD}_{+/-}$ model $\mathscr{D}'$ is ε-close to $\mathbf{P}$ as required. □

By combining Theorem 5.7 and Lemma 5.9 we obtain the following theorem.

Theorem 5.10. *Let* $\mathbf{C}$ *be closed under removing tuples, removing elements and inserting elements. If a property* $\mathbf{P}\subseteq\mathbf{C}$ *is hyperfinite on* $\mathbf{C}$*, then* $\mathbf{P}$ *is local on* $\mathbf{C}$ *in the* $\mathrm{BDRD}_{+/-}$ *model.*

5.3 Constant time testability of monadic second-order logic with counting

We begin this section with the first of our main theorems (Theorem 5.11). We show that for any property $\mathbf{P}$ which is ε-local (in the $\mathrm{BDRD}_{+/-}$ model) on the input class $\mathbf{C}$, if the set of r-histograms of $\mathbf{P}$ is semilinear, then for every σ-db $\mathscr{D}$ in $\mathbf{P}$ there exists a constant size σ-db in $\mathbf{P}$ with a neighbourhood distribution similar to that of $\mathscr{D}$, but this is not true for σ-dbs in $\mathbf{C}$ that are far from $\mathbf{P}$. We then use this result to prove that for such properties there exist ε-testers in the $\mathrm{BDRD}_{+/-}$ model that run in constant time (Theorem 5.13). As corollaries we obtain that hyperfinite properties whose set of r-histograms is semilinear is constant time

testable (Theorem 5.14) and CMSO definable properties on σ-dbs of bounded tree-width and bounded degree are uniformly testable in constant time (Theorem 5.15).

Theorem 5.11. *Let $\varepsilon \in (0,1]$. Let $\mathbf{P} \subseteq \mathbf{C}$ be a property that is ε-local on $\mathbf{C}$ (in the $\mathrm{BDRD}_{+/-}$ model) such that the set $\mathrm{h}_r(\mathbf{P})$ is semilinear, where $r := r_{5.8}(\varepsilon)$ is the locality radius of $\mathbf{P}$ for ε. Then there exist $n_{min} := n_{min}(\varepsilon), n_{max} := n_{max}(\varepsilon) \in \mathbb{N}$ and $f := f(\varepsilon), \mu := \mu(\varepsilon) \in (0,1)$ such that for every $\mathscr{D} \in \mathbf{C}$ with $|D| > n_{max}$,*

1. *if $\mathscr{D} \in \mathbf{P}$, then there exists a $\mathscr{D}' \in \mathbf{P}$ such that $n_{min} \leq |D'| \leq n_{max}$ and $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f - \mu$, and*
2. *if $\mathscr{D}$ is ε-far from $\mathbf{P}$ (in the $\mathrm{BDRD}_{+/-}$ model), then for every $\mathscr{D}' \in \mathbf{P}$ such that $n_{min} \leq |D'| \leq n_{max}$, we have $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 > f + \mu$.*

Proof. Let $\lambda := \lambda_{5.8}(\varepsilon)$ and $N := N_{5.8}(\varepsilon)$ be as in Definition 5.8 for $\mathbf{P}$ and ε, and let $c := \mathrm{c}(r)$ (the number of r-types with one centre). First note that if $\mathbf{P}$ is empty then for any choice of $n_{\min}$, $n_{\max}$, f and μ, both 1 and 2 in the theorem statement are true and hence we shall assume that $\mathbf{P}$ is non-empty. As $\mathrm{h}_r(\mathbf{P})$ is a semilinear set we can write it as follows, $\mathrm{h}_r(\mathbf{P}) = M_1 \cup M_2 \cup \cdots \cup M_m$ where $m \in \mathbb{N}$ and for each $i \in [m]$, $M_i = \{\vec{v}_0^i + a_1\vec{v}_1^i + \cdots + a_{k_i}\vec{v}_{k_i}^i \mid a_1, \ldots, a_{k_i} \in \mathbb{N}\}$ is a linear set where $\vec{v}_0^i, \ldots, \vec{v}_{k_i}^i \in \mathbb{N}^c$ and for each $j \in [k_i]$, $\|\vec{v}_j^i\|_1 \neq 0$. Let $k := \max_{i\in[m]} k_i + 1$ and $v := \max_{i\in[m]} \left(\max_{j\in[0,k_i]} \|\vec{v}_j^i\|_1\right)$ (note that $v > 0$ as $\mathbf{P}$ is non-empty). Let

- $n_{\min} := n_0 - kv$,
- $n_{\max} := n_0 + kv$,
- $f := \frac{\lambda}{3c}$, and
- $\mu := \frac{\lambda}{6c}$

where

$$n_0 := \max\left\{\frac{9N}{5}, kv\left(\frac{3ckv}{f-\mu} + 1\right)\right\}.$$

Note that $n_{\min} > 0$ by the choice of n_0, f and μ.

(Proof of 1.) Assume $\mathscr{D} \in \mathbf{P}$ and $|D| = n > n_{\max}$. Then by Theorem 4.2 there exists a $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$ and $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f - \mu$.

(Proof of 2.) Assume $\mathscr{D}$ is ε-far from $\mathbf{P}$ and $|D| = n > n_{\max}$. For a contradiction let us assume there does exist a σ-db $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$ and $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f + \mu$. Since $\mathscr{D}' \in \mathbf{P}$ there exists some $i \in [m]$ and $a_1^{\mathscr{D}'}, \ldots, a_{k_i}^{\mathscr{D}'} \in \mathbb{N}$ such that $\mathrm{h}_r(\mathscr{D}') = \vec{v}_0^i + a_1^{\mathscr{D}'}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}'}\vec{v}_{k_i}^i$. Let $\mathscr{D}''$ be the σ-db with r-histogram $\vec{v}_0^i + a_1^{\mathscr{D}''}\vec{v}_1^i + \cdots +$

$a^{\mathscr{D}''}_{k_i}\vec{v}^i_{k_i} \in M_i$ where $a^{\mathscr{D}''}_j$ is the nearest integer to $a^{\mathscr{D}'}_j n/|D'|$. Note as $\vec{v}^i_0 + a^{\mathscr{D}''}_1\vec{v}^i_1 + \cdots + a^{\mathscr{D}''}_{k_i}\vec{v}^i_{k_i} \in \mathrm{h}_r(\mathbf{P})$, $\mathscr{D}''$ exists and $\mathscr{D}'' \in \mathbf{P}$.

Claim 5.12. *$\mathscr{D}$ is ε-close to* $\mathbf{P}$.

Proof: Since $\mathbf{P}$ is ε-local on $\mathbf{C}$ (and $|D| \geq N$, $\mathscr{D} \in \mathbf{C}$ and $\mathscr{D}'' \in \mathbf{P}$), if $|D''| \geq N$ and $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}'')\|_1 \leq \lambda \min\{n, |D''|\}$ then $\mathscr{D}$ is ε-close to $\mathbf{P}$. We will start by obtaining a bound on $|\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}'')[j]|$. First note that as $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f + \mu$ and $\mathrm{h}_r(\mathscr{D}') = \vec{v}^i_0 + a^{\mathscr{D}'}_1\vec{v}^i_1 + \cdots + a^{\mathscr{D}'}_{k_i}\vec{v}^i_{k_i}$, for every $j \in [c]$

$$\frac{\vec{v}^i_0[j] + \sum_{\ell\in[k_i]} a^{\mathscr{D}'}_\ell \vec{v}^i_\ell[j]}{|D'|} - f - \mu \leq \mathrm{dv}_r(\mathscr{D})[j] \leq \frac{\vec{v}^i_0[j] + \sum_{\ell\in[k_i]} a^{\mathscr{D}'}_\ell \vec{v}^i_\ell[j]}{|D'|} + f + \mu$$

and therefore

$$n\Big(\frac{\vec{v}^i_0[j] + \sum_{\ell\in[k_i]} a^{\mathscr{D}'}_\ell \vec{v}^i_\ell[j]}{|D'|} - f - \mu\Big) \leq \mathrm{h}_r(\mathscr{D})[j] \leq n\Big(\frac{\vec{v}^i_0[j] + \sum_{\ell\in[k_i]} a^{\mathscr{D}'}_\ell \vec{v}^i_\ell[j]}{|D'|} + f + \mu\Big).$$

Hence, by the choice of $a^{\mathscr{D}''}_\ell$ for $\ell \in [k_i]$,

$$\begin{aligned}
\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}'')[j] &\leq \vec{v}^i_0[j]\Big(\frac{n}{|D'|} - 1\Big) + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} - a^{\mathscr{D}''}_\ell\Big) + fn + \mu n \\
&\leq \vec{v}^i_0[j]\frac{n}{|D'|} + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} - \Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} - \frac{1}{2}\Big)\Big) + fn + \mu n \\
&= \vec{v}^i_0[j]\frac{n}{|D'|} + \frac{1}{2}\sum_{\ell\in[k_i]} \vec{v}^i_\ell[j] + fn + \mu n.
\end{aligned}$$

Similarly, by the choice of $a^{\mathscr{D}''}_\ell$ for $\ell \in [k_i]$ and as $n > |D'|$,

$$\begin{aligned}
\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}'')[j] &\geq \vec{v}^i_0[j]\Big(\frac{n}{|D'|} - 1\Big) + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} - a^{\mathscr{D}''}_\ell\Big) - fn - \mu n \\
&\geq -\vec{v}^i_0[j]\frac{n}{|D'|} + \sum_{\ell\in[k_i]} \vec{v}^i_\ell[j]\Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} - \Big(\frac{a^{\mathscr{D}'}_\ell n}{|D'|} + \frac{1}{2}\Big)\Big) - fn - \mu n \\
&= -\vec{v}^i_0[j]\frac{n}{|D'|} - \frac{1}{2}\sum_{\ell\in[k_i]} \vec{v}^i_\ell[j] - fn - \mu n.
\end{aligned}$$

Therefore,

$$\begin{aligned}
|\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}'')[j]| &\leq \vec{v}_0^i[j]\frac{n}{|D'|} + \frac{1}{2}\sum_{\ell\in[k_i]} \vec{v}_\ell^i[j] + fn + \mu n \\
&\leq \frac{n}{|D'|}\sum_{0\leq\ell\leq k_i} \vec{v}_\ell^i[j] + fn + \mu n \\
&\leq \frac{nkv}{|D'|} + fn + \mu n \\
&= n\Big(\frac{kv}{|D'|} + \frac{\lambda}{3c} + \frac{\lambda}{6c}\Big) \\
&\leq n\Big(\frac{\lambda}{18c} + \frac{\lambda}{3c} + \frac{\lambda}{6c}\Big) \\
&= \frac{5\lambda n}{9c}
\end{aligned}$$

by the choice of f and μ and since

$$|D'| \geq n_{\min} \geq \frac{3c(kv)^2}{f-\mu} = \frac{18(ckv)^2}{\lambda} \geq \frac{18ckv}{\lambda}.$$

If $|\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}'')[j]| \leq \frac{\lambda}{c}\min\{n, |D''|\}$ then $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}'')\|_1 \leq \lambda\min\{n, |D''|\}$. Clearly, $\frac{5\lambda n}{9c} < \frac{\lambda n}{c}$, but we must also show that $\frac{5\lambda n}{9c} \leq \frac{\lambda|D''|}{c}$. We have

$$\begin{aligned}
|D''| &= \|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}''}\|\vec{v}_\ell^i\|_1 \\
&\geq \|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} \Big(\frac{a_\ell^{\mathscr{D}'}n}{|D'|} - \frac{1}{2}\Big)\|\vec{v}_\ell^i\|_1 \\
&= \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{\ell\in[k_i]} \|\vec{v}_\ell^i\|_1 + \frac{n}{|D'|}\sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'}\|\vec{v}_\ell^i\|_1 \\
&\geq -kv + \frac{n}{|D'|}(|D'| - \|\vec{v}_0^i\|_1) \\
&\geq -\frac{n}{18} + \frac{17}{18}n \\
&> \frac{5n}{9}
\end{aligned}$$

since

$$|D'| \geq \frac{18ckv}{\lambda} \geq 18v \geq 18\|\vec{v}_0^i\|_1 \text{ and } kv \leq \frac{(ckv)^2}{\lambda} \leq \frac{n_{\min}}{18} \leq \frac{n}{18}.$$

Therefore, $\frac{5\lambda n}{9c} \leq \frac{\lambda |D''|}{c}$ and hence $\| \mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}'')\|_1 \leq \lambda \min\{n, |D''|\}$. Furthermore, by the choice of $n_{\max}$, $\frac{5n}{9} \geq N$ and hence $|D''| \geq N$. Therefore, $\mathscr{D}$ is ε-close to $\mathbf{P}$. ■

Claim 5.12 gives us a contradiction and therefore for every $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$, we have $\| \mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 > f + \mu$ as required. □

Our aim is to construct constant time testers for local properties whose set of r-histograms are semilinear. If we can approximate the (one-centre) r-neighbourhood distribution of a σ-db then by Theorem 5.11 we only need to check whether this distribution is close or not to the r-neighbourhood distribution of some small constant size σ-db. Recall that $\mathrm{EstimateFrequencies}_{r,k,s}$ is the algorithm that, given oracle access to an input σ-db $\mathscr{D}$, samples s many elements uniformly and independently from D^k and computes their r-type. The algorithm then returns the k-centre r-neighbourhood distribution vector of the sample.

Theorem 5.13. *Let $\varepsilon \in (0,1]$ and let $\mathbf{P} \subseteq \mathbf{C}$ be a property that is ε-local on $\mathbf{C}$ (in the $\mathrm{BDRD}_{+/-}$ model). If for each $r \in \mathbb{N}$ the set $\mathrm{h}_r(\mathbf{P})$ is semilinear, then there exists an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ in the $\mathrm{BDRD}_{+/-}$ model that has constant running time and constant query complexity.*

Proof. Let $r := r_{5.8}(\varepsilon)$ be the locality radius of $\mathbf{P}$ for ε, let $n_{\min} := n_{\min}(\varepsilon)$, $n_{\max} := n_{\max}(\varepsilon)$, $f := f(\varepsilon)$ and $\mu := \mu(\varepsilon)$ be as in Theorem 5.11 and let $s = \mathrm{c}(r)^2/\mu^2 \cdot \ln(20\,\mathrm{c}(r))$. Assume that the set $\mathrm{h}_r(\mathbf{P})$ is semilinear. Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}$ and $|D| = n$ as an input, the ε-tester for $\mathbf{P}$ on $\mathbf{C}$ proceeds as follows:

1. If $n \leq n_{\max}$, do a full check of $\mathscr{D}$ and decide if $\mathscr{D} \in \mathbf{P}$.
2. Run $\mathrm{EstimateFrequencies}_{r,1,s}$ and let $\bar{\nu}$ be the resulting vector.
3. If there exists a $\mathscr{D}' \in \mathbf{P}$ where $n_{\min} \leq |D'| \leq n_{\max}$ and $\|\bar{\nu} - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f$ then accept, otherwise reject.

The running time and query complexity of the above tester is constant as $n_{\max}$ is a constant (it only depends on $\mathbf{P}$, d, σ and ε) and $\mathrm{EstimateFrequencies}_{r,1,s}$ runs in constant time and makes a constant number of oracle queries.

For correctness, first assume $\mathscr{D} \in \mathbf{P}$. By Theorem 5.11 there exists a σ-db $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$ and $\| \mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f - \mu$. By Lemma 2.19 with probability at least $9/10$, $\|\bar{\nu} - \mathrm{dv}_r(\mathscr{D})\|_1 \leq \mu$ and therefore $\|\bar{\nu} - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f$. Hence with probability at least $9/10$ the tester will accept.

Now assume $\mathscr{D}$ is ε-far from $\mathbf{P}$. By Theorem 5.11 for every $\mathscr{D}' \in \mathbf{P}$ with $n_{\min} \leq |D'| \leq n_{\max}$, we have $\| \mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 > f + \mu$. By Lemma 2.19 with probability at least $9/10$, $\|\bar{\nu} - \mathrm{dv}_r(\mathscr{D})\|_1 \leq \mu$ and therefore for every $\mathscr{D}' \in \mathbf{P}$ with $n_{\min} \leq |D'| \leq n_{\max}$, $\|\bar{\nu} - \mathrm{dv}_r(\mathscr{D}')\|_1 > f$. Hence with probability at least $9/10$ the tester will reject. □

Combining Theorems 5.10 and 5.13 we obtain the following as a corollary.

Theorem 5.14. *Let* $\mathbf{C}$ *be closed under removing tuples, removing elements and inserting elements. Let* $\mathbf{P} \subseteq \mathbf{C}$ *be a property that is hyperfinite on* $\mathbf{C}$*. If for each* $r \in \mathbb{N}$ *the set* $\mathrm{h}_r(\mathbf{P})$ *is semilinear, then* $\mathbf{P}$ *is uniformly testable on* $\mathbf{C}$ *in constant time in the* $\mathrm{BDRD}_{+/-}$ *model.*

Let $\mathscr{G}$ be a graph. It is known that the tree-width of any subgraph of $\mathscr{G}$ is at most the tree-width of $\mathscr{G}$ (e.g. see [19] for a proof). Furthermore, the tree-width of a graph is the maximum tree-width of its connected components (e.g. see [19] for a proof). Hence since the tree-width of a database is the tree-width of its Gaifman graph, the class of databases $\mathbf{C}_d^t$ is closed under removing tuples, removing elements and inserting (isolated) elements. The class $\mathbf{C}_d^t$ is hyperfinite [45, 9] (and so any property is hyperfinite on $\mathbf{C}_d^t$) and so by combining Theorem 5.14 and Lemma 2.40 we obtain the following as a corollary.

Theorem 5.15. *Every property* $\mathbf{P}$ *definable by a CMSO sentence on* $\mathbf{C}_d^t$ *is uniformly testable on* $\mathbf{C}_d^t$ *with constant time complexity in the* $\mathrm{BDRD}_{+/-}$ *model.*

5.4 Hyperfinitness and near semilinearity together implies constant time testability

We begin this section by defining the notion of δ-indistinguishability, which is based on the definition of indistinguishability in the dense graph model given in [6]. We then prove that any hyperfinite property which, for every $\delta \in (0,1]$, is δ-indistinguishable from a property whose r-histograms are semilinear is constant time testable in the $\mathrm{BDRD}_{+/-}$ model (Theorem 5.19).

Definition 5.16 (δ-indistinguishable)**.** *Let* $\delta \in (0,1]$*. Two properties* $\mathbf{P}$ *and* $\mathbf{Q}$ *are called* δ-indistinguishable *if there exists* $N := N_{5.16}(\delta) \in \mathbb{N}$ *that satisfies the following. For every* σ*-db* $\mathscr{D} \in \mathbf{P}$ *with* $|D| = n \geq N$ *elements there exists a* σ*-db* $\mathscr{D}' \in \mathbf{Q}$ *such that* $\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \delta d \min\{n, |D'|\}$*; and for every* σ*-db* $\mathscr{D} \in \mathbf{Q}$ *with* $|D| = n \geq N$ *elements there exists a* σ*-db* $\mathscr{D}' \in \mathbf{P}$ *such that* $\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \delta d \min\{n, |D'|\}$.

The following two lemmas will be useful in the proof of Theorem 5.19.

Lemma 5.17. *Let* $\mathscr{D}$ *and* $\mathscr{D}'$ *be two* σ*-dbs and let* $\delta \in (0,1]$*. If*

$$\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \delta d \min\{|D|, |D'|\}$$

then for any $r \in \mathbb{N}$*,*

$$\| \mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}') \|_1 \leq 3c\delta d^{r+2} \mathrm{ar}(\sigma)^{r+1}$$

and

$$\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\min\{|D|,|D'|\}$$

where $c := c(r)$.

Proof. Let $\delta \in (0,1]$ and let $r \in \mathbb{N}$. Let us assume that $\mathrm{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \delta d \min\{|D|,|D'|\}$. The distance between $\mathscr{D}$ and $\mathscr{D}'$ is the minimum number of modifications needed to make $\mathscr{D}$ and $\mathscr{D}'$ isomorphic. The four different types of modifications allowed are: (1) inserting a new element, (2) deleting an element, (3) inserting a new tuple, and (4) deleting a tuple. If a new element is added to $\mathscr{D}$ or $\mathscr{D}'$ then no existing elements r-type is changed. If an element is deleted from $\mathscr{D}$ or $\mathscr{D}'$ then the r-type of any element at distance at most r from the deleted element could have changed. If a tuple $\bar{a}$ is inserted or deleted from $\mathscr{D}$ or $\mathscr{D}'$ then the r-type of any element that is at distance at most r from every element in $\bar{a}$ could have changed. Hence, since the number of elements in the r-neighbourhood of an element is at most $(\mathrm{ar}(\sigma)\cdot d)^{r+1}$ (by Lemma 2.16), every modification to $\mathscr{D}$ or $\mathscr{D}'$ could change the r-type of at most $(\mathrm{ar}(\sigma)\cdot d)^{r+1}$ many elements. Therefore, $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\min\{|D|,|D'|\}$ as required.

By definition,

$$\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 = \sum_{i=1}^{c}\Big|\frac{\mathrm{h}_r(\mathscr{D})[i]}{|D|} - \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D'|}\Big|.$$

Let $i \in [c]$, then since $\|\mathrm{h}_r(\mathscr{D}) - \mathrm{h}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\min\{|D|,|D'|\}$,

$$\begin{aligned}\frac{\mathrm{h}_r(\mathscr{D})[i]}{|D|} - \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D'|} &\leq \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D|} + \frac{2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\min\{|D|,|D'|\}\}}{|D|} - \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D'|}\\ &\leq \mathrm{h}_r(\mathscr{D}')[i]\Big(\frac{1}{|D|} - \frac{1}{|D'|}\Big) + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}.\end{aligned}$$

Then since $|D'| \leq |D|(1+\delta d)$ (as $\mathrm{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \delta d\min\{|D|,|D'|\}$) we have $|D| \geq |D'|/(1+\delta d)$ and hence

$$\mathrm{h}_r(\mathscr{D}')[i]\Big(\frac{1}{|D|} - \frac{1}{|D'|}\Big) + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1} \leq \frac{\mathrm{h}_r(\mathscr{D}')[i]\delta d}{|D'|} + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}.$$

Similarly,

$$\begin{aligned}\frac{\mathrm{h}_r(\mathscr{D})[i]}{|D|} - \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D'|} &\geq \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D|} - \frac{2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\min\{|D|,|D'|\}}{|D|} - \frac{\mathrm{h}_r(\mathscr{D}')[i]}{|D'|}\\ &\geq \mathrm{h}_r(\mathscr{D}')[i]\Big(\frac{1}{|D|} - \frac{1}{|D'|}\Big) - 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\end{aligned}$$

$$\geq \frac{-\mathrm{h}_r(\mathscr{D}')[i]\delta d}{|D'|} - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$$

since $|D'| \geq |D|(1-\delta d)$ and hence $|D| \leq |D'|/(1-\delta d)$. Hence,

$$\begin{aligned}\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 &\leq \sum_{i=1}^{c}\Big(\frac{\mathrm{h}_r(\mathscr{D}')[i]\delta d}{|D'|} + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}\Big)\\ &= \delta d + 2c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}\\ &\leq 3c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}\end{aligned}$$

as required. □

Lemma 5.18. *Let $\mathscr{D}_1, \mathscr{D}_2, \mathscr{D}_3 \in \mathbf{C}$, let $r \in \mathbb{N}$ and let $x \in \mathbb{R}$ such that $x \geq 0$. If $\big||D_1| - |D_2|\big| \leq x\min\{|D_1|, |D_2|\}$, then*

$$(1-x)\min\{|D_1|, |D_3|\} \leq \min\{|D_2|, |D_3|\} \leq (1+x)\min\{|D_1|, |D_3|\}.$$

Proof. Let us assume that $\big||D_1| - |D_2|\big| \leq x\min\{|D_1|, |D_2|\}$. To prove that $\min\{|D_2|, |D_3|\} \leq (1+x)\min\{|D_1|, |D_3|\}$ we shall consider the cases where out of $|D_1|$, $|D_2|$ and $|D_3|$, (1) $|D_1|$ is the smallest, (2) $|D_2|$ is the smallest, and (3) $|D_3|$ is the smallest. In cases (2) and (3), since $(1+x) \geq 1$, the inequality holds. For case (1),

$$\min\{|D_2|, |D_3|\} \leq |D_2| \leq |D_1| + x\min\{|D_1|, |D_2|\} = (1+x)\min\{|D_1|, |D_3|\}$$

since $\big||D_1| - |D_2|\big| \leq x\min\{|D_1|, |D_2|\}$ and $|D_1|$ is the smallest.

To prove that $(1-x)\min\{|D_1|, |D_3|\} \leq \min\{|D_2|, |D_3|\}$ we shall again consider the above three cases. In cases (1) and (3), since $1-x \leq 1$, the inequality holds. For case (2),

$$\min\{|D_2|, |D_3|\} = |D_2| \geq (1-x)|D_1| \geq (1-x)\min\{|D_1|, |D_3|\}$$

since $\big||D_1| - |D_2|\big| \leq x\min\{|D_1|, |D_2|\} \leq x|D_1|$. □

We now prove our main result of this section.

Theorem 5.19. *Let $\mathbf{C}$ be closed under removing tuples, removing elements and inserting elements. Let $\mathbf{P} \subseteq \mathbf{C}$ be a property that is hyperfinite on $\mathbf{C}$. If for every $\delta \in (0,1]$ there exists a property $\mathbf{Q}_\delta \subseteq \mathbf{C}$ such that*

1. *$\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable, and*
2. *for every $r \in \mathbb{N}$, $\mathrm{h}_r(\mathbf{Q}_\delta)$ is semilinear*

then $\mathbf{P}$ *is uniformly testable on* $\mathbf{C}$ *in constant time.*

Proof. Let $\varepsilon \in (0,1]$. We shall prove that there exists an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ that runs in constant time and has constant query complexity. By Theorem 5.10, $\mathbf{P}$ is local on $\mathbf{C}$ in the $\mathrm{BDRD}_{+/-}$ model. Let $N_{\mathbf{P}} := N_{5.8}(\varepsilon/4)$, $\lambda_{\mathbf{P}} := \lambda_{5.8}(\varepsilon/4)$ and $r_{\mathbf{P}} := r_{5.8}(\varepsilon/4)$ be as in Definition 5.8 for $\mathbf{P}$ and $\varepsilon/4$. Let $c := c(r_{\mathbf{P}})$. Let

$$\delta := \min\left\{\frac{\varepsilon}{6d}, \frac{\lambda_{\mathbf{P}}}{40c^2 d^{r+2}\operatorname{ar}(\sigma)^{r+1}}\right\}$$

and let us assume that there exists a property $\mathbf{Q}_\delta \subseteq \mathbf{C}$ that is δ-indistinguishable from $\mathbf{P}$ and for every $r \in \mathbb{N}$, $\mathrm{h}_r(\mathbf{Q}_\delta)$ is semilinear.

Let $r := r_{\mathbf{P}}$, $\lambda := \lambda_{\mathbf{P}}/2$ and $N := (N_{\mathbf{P}}+1)(N_{5.16}(\delta)+1)(1+\varepsilon d/4)$ where $N_{5.16}(\delta)$ is as in Definition 5.16 for $\mathbf{P}$ and $\mathbf{Q}_\delta$.

Claim 5.20. *The property* $\mathbf{Q}_\delta$ *is* $\varepsilon/2$*-local on* $\mathbf{C}$ *in the* $\mathrm{BDRD}_{+/-}$ *model with* $r_{5.8}(\varepsilon/2) = r$, $\lambda_{5.8}(\varepsilon/2) = \lambda$ *and* $N_{5.8}(\varepsilon/2) = N$.

Proof: Let $\mathscr{D}_{\mathbf{Q}} \in \mathbf{Q}_\delta$ and let $\mathscr{D}_{\mathbf{C}} \in \mathbf{C}$ such that $|D_{\mathbf{Q}}| \geq N$ and $|D_{\mathbf{C}}| \geq N$. Let us assume that $\|\mathrm{h}_r(\mathscr{D}_{\mathbf{Q}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{C}})\|_1 \leq \lambda \min\{|D_{\mathbf{Q}}|, |D_{\mathbf{C}}|\}$. We need to prove that $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/2$-close to $\mathbf{Q}_\delta$.

We will start by showing that $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/4$-close to $\mathbf{P}$ (using the locality of $\mathbf{P}$ on $\mathbf{C}$). Since $\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable and $N \geq N_{5.16}(\delta)$, there exists $\mathscr{D}_{\mathbf{P}} \in \mathbf{P}$ such that $\mathrm{dist}_{+/-}(\mathscr{D}_{\mathbf{P}}, \mathscr{D}_{\mathbf{Q}}) \leq \delta d \min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}$. By Lemma 5.17,

$$\|\mathrm{h}_r(\mathscr{D}_{\mathbf{P}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{Q}})\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}\min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}.$$

Hence,

$$\|\mathrm{h}_r(\mathscr{D}_{\mathbf{P}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{C}})\|_1 \leq \lambda \min\{|D_{\mathbf{Q}}|, |D_{\mathbf{C}}|\} + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}\min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}.$$

We have

$$\big||D_{\mathbf{P}}| - |D_{\mathbf{Q}}|\big| \leq \delta d \min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}$$

since $\mathrm{dist}_{+/-}(\mathscr{D}_{\mathbf{P}}, \mathscr{D}_{\mathbf{Q}}) \leq \delta d \min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}$ and we have

$$\big||D_{\mathbf{C}}| - |D_{\mathbf{Q}}|\big| \leq \lambda \min\{|D_{\mathbf{C}}|, |D_{\mathbf{Q}}|\}$$

as $\|\mathrm{h}_r(\mathscr{D}_{\mathbf{Q}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{C}})\|_1 \leq \lambda \min\{|D_{\mathbf{Q}}|, |D_{\mathbf{C}}|\}$. Hence by Lemma 5.18,

$$\min\{|D_{\mathbf{Q}}|, |D_{\mathbf{C}}|\} \leq (1+\delta d)\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\}$$

(where $\mathscr{D}_1 = \mathscr{D}_{\mathbf{P}}$, $\mathscr{D}_2 = \mathscr{D}_{\mathbf{Q}}$ and $\mathscr{D}_3 = \mathscr{D}_{\mathbf{C}}$) and

$$\min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\} \leq (1+\lambda)\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\}$$

(where $\mathscr{D}_1 = \mathscr{D}_{\mathbf{C}}$, $\mathscr{D}_2 = \mathscr{D}_{\mathbf{Q}}$ and $\mathscr{D}_3 = \mathscr{D}_{\mathbf{P}}$).

Therefore,

$$\begin{aligned}
&\|\mathrm{h}_r(\mathscr{D}_{\mathbf{P}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{C}})\|_1 \\
&\leq \lambda(1+\delta d)\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\} + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}(1+\lambda)\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\} \\
&\leq (\lambda + 5\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1})\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\} \\
&\leq \lambda_{\mathbf{P}}\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\}
\end{aligned}$$

by the choice of λ and δ. Since $\mathrm{dist}_{+/-}(\mathscr{D}_{\mathbf{P}}, \mathscr{D}_{\mathbf{Q}}) \leq \delta d \min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\}$ we have $|D_{\mathbf{P}}| \geq |D_{\mathbf{Q}}| - \delta d \min\{|D_{\mathbf{P}}|, |D_{\mathbf{Q}}|\} \geq |D_{\mathbf{Q}}| - \delta d |D_{\mathbf{P}}|$. Hence,

$$|D_{\mathbf{P}}| \geq \frac{|D_{\mathbf{Q}}|}{1+\delta d} \geq \frac{N}{1+\varepsilon d/4} \geq N_{\mathbf{P}}$$

by the choice of δ and N. Therefore, since $\mathbf{P}$ is local on $\mathbf{C}$ (and $r = r_{\mathbf{P}}$, $|D_{\mathbf{P}}| \geq N_{\mathbf{P}}$, $|D_{\mathbf{C}}| \geq N \geq N_{\mathbf{P}}$ and $\|\mathrm{h}_r(\mathscr{D}_{\mathbf{P}}) - \mathrm{h}_r(\mathscr{D}_{\mathbf{C}})\|_1 \leq \lambda_{\mathbf{P}}\min\{|D_{\mathbf{P}}|, |D_{\mathbf{C}}|\}$), $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/4$-close to $\mathbf{P}$.

We will now show that since $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/4$-close to $\mathbf{P}$, $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/2$-close to $\mathbf{Q}_\delta$. Since $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/4$-close to $\mathbf{P}$, there exists $\mathscr{D}'_{\mathbf{P}} \in \mathbf{P}$ such that

$$\mathrm{dist}_{+/-}(\mathscr{D}_{\mathbf{C}}, \mathscr{D}'_{\mathbf{P}}) \leq \frac{\varepsilon d \min\{|D_{\mathbf{C}}|, |D'_{\mathbf{P}}|\}}{4}$$

which implies $\big||D_{\mathbf{C}}| - |D'_{\mathbf{P}}|\big| \leq \varepsilon d \min\{|D_{\mathbf{C}}|, |D'_{\mathbf{P}}|\}/4$. Therefore

$$|D'_{\mathbf{P}}| \geq |D_{\mathbf{C}}| - \frac{\varepsilon d \min\{|D_{\mathbf{C}}|, |D'_{\mathbf{P}}|\}}{4} \geq |D_{\mathbf{C}}| - \frac{\varepsilon d |D'_{\mathbf{P}}|}{4},$$

and hence we have

$$|D'_{\mathbf{P}}| \geq \frac{|D_{\mathbf{C}}|}{(1+\varepsilon d/4)} \geq \frac{N}{(1+\varepsilon d/4)} \geq N_{5.16}(\delta)$$

by the choice of N. Therefore there exists $\mathscr{D}'_{\mathbf{Q}} \in \mathbf{Q}_\delta$ such that $\mathrm{dist}_{+/-}(\mathscr{D}'_{\mathbf{P}}, \mathscr{D}'_{\mathbf{Q}}) \leq \delta d \min\{|D'_{\mathbf{P}}|, |D'_{\mathbf{Q}}|\}$ (which implies $\big||D'_{\mathbf{Q}}| - |D'_{\mathbf{P}}|\big| \leq \delta d \min\{|D'_{\mathbf{Q}}|, |D'_{\mathbf{P}}|\}$) . Hence,

$$\mathrm{dist}_{+/-}(\mathscr{D}_{\mathbf{C}}, \mathscr{D}'_{\mathbf{Q}}) \leq \frac{\varepsilon d \min\{|D_{\mathbf{C}}|, |D'_{\mathbf{P}}|\}}{4} + \delta d \min\{|D'_{\mathbf{P}}|, |D'_{\mathbf{Q}}|\}$$

$$
\begin{aligned}
&\leq \frac{\varepsilon d(1+\delta d)\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}}{4} + \delta d\Big(1+\frac{\varepsilon d}{4}\Big)\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}\\
&= \Big(\frac{\varepsilon d}{4} + \frac{\varepsilon\delta d^2}{2} + \delta d\Big)\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}\\
&\leq \Big(\frac{\varepsilon d}{4} + \frac{\varepsilon^2 d}{12} + \frac{\varepsilon}{6}\Big)\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}\\
&\leq \Big(\frac{\varepsilon d}{4} + \frac{\varepsilon d}{12} + \frac{\varepsilon d}{6}\Big)\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}\\
&= \frac{\varepsilon d}{2}\min\{|D_{\mathbf{C}}|,|D'_{\mathbf{Q}}|\}
\end{aligned}
$$

by Lemma 5.18 and the choice of δ. Therefore, $\mathscr{D}_{\mathbf{C}}$ is $\varepsilon/2$-close to $\mathbf{Q}_\delta$ as required. ■

By Claim 5.20 and Theorem 5.13 there exists an $\varepsilon/2$-tester for $\mathbf{Q}_\delta$ on $\mathbf{C}$ that runs in constant time and has constant query complexity. Let $\mu := \mu(\varepsilon/2)$, $f := f(\varepsilon/2)$, $n_{\min} := n_{\min}(\varepsilon/2)$ and $n_{\max} := n_{\max}(\varepsilon/2)$ be as in Theorem 5.11 for $\mathbf{Q}_\delta$ and $\varepsilon/2$. Note that by Claim 5.20 the locality radius and disc proximity of $\mathbf{Q}_\delta$ for $\varepsilon/2$ are $r_{\mathbf{P}}$ and $\lambda_{\mathbf{P}}/2$ respectively, and therefore by Theorem 5.11, $\mu = \lambda_{\mathbf{P}}/12c$. Let $\pi_{\varepsilon/2}$ be the $\varepsilon/2$-tester for $\mathbf{Q}_\delta$ on $\mathbf{C}$ from the proof of Theorem 5.13 but in $\pi_{\varepsilon/2}$ let us increase the number of elements sampled in the second step to $s = \mathrm{c}^2/(\mu - 3c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1})^2 \cdot \ln(20\,\mathrm{c})$. Note that $\mu - 3c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1} \in (0,1)$ by the choice of δ and since $\mu = \lambda_{\mathbf{P}}/12c$. Then Given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}$ and $|D| = n$ as an input, the ε-tester for $\mathbf{P}$ on $\mathbf{C}$ proceeds as follows.

1. If $n < N_{5.16}(\delta)(1+\varepsilon d/2)$, do a full check of $\mathscr{D}$ and decide if $\mathscr{D} \in \mathbf{P}$.
2. Run $\pi_{\varepsilon/2}$ on $\mathscr{D}$ and accept if $\pi_{\varepsilon/2}$ accepts and reject otherwise.

Clearly the above tester runs in constant time and has constant query complexity.

For correctness, first assume that $\mathscr{D} \in \mathbf{P}$, $|D| = n > n_{\max}$ and $n \geq N_{5.16}(\delta)(1+\varepsilon d/2)$ (otherwise the tester will accept with probability 1). As $\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable there exists a σ-db $\mathscr{D}' \in \mathbf{Q}_\delta$ such that

$$\operatorname{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \delta d\min\{n,|D'|\}.$$

By Lemma 5.17,

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\| \leq 3c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}.$$

By Theorem 5.11 there exists a σ-db $\mathscr{D}_0 \in \mathbf{Q}_\delta$ such that $n_{\min} \leq |D_0| \leq n_{\max}$ and $\|\operatorname{dv}_r(\mathscr{D}') - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq f - \mu$. Hence

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_0)\|_1 \leq f - \mu + 3c\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}.$$

By Lemma 2.19 and the choice of s with probability at least $9/10$, the vector $\bar{v}$ returned in Step 2 of $\pi_{\varepsilon/2}$ satisfies

$$\|\bar{v} - \mathrm{dv}_r(\mathscr{D})\|_1 \leq \mu - 3c\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}$$

and therefore $\|\bar{v} - \mathrm{dv}_r(\mathscr{D}_0)\|_1 \leq f$. Hence with probability at least $9/10$ the tester will accept.

Now assume $\mathscr{D}$ is ε-far from $\mathbf{P}$ and $|D| \geq N_{5.16}(\delta)(1+\varepsilon d/2)$. We will show that $\mathscr{D}$ is $\varepsilon/2$-far from $\mathbf{Q}_\delta$. Let $\mathscr{D}_\mathbf{Q} \in \mathbf{Q}_\delta$. If $\big||D_\mathbf{Q}| - |D|\big| > \varepsilon d \min\{|D_\mathbf{Q}|, |D|\}/2$, then $\mathscr{D}$ is $\varepsilon/2$-far from $\mathscr{D}_\mathbf{Q}$. So let us assume that $\big||D_\mathbf{Q}| - |D|\big| \leq \varepsilon d \min\{|D_\mathbf{Q}|, |D|\}/2$. This implies that

$$\frac{|D|}{1+\varepsilon d/2} \leq |D_\mathbf{Q}| \leq |D|(1+\varepsilon d/2).$$

Hence, since $|D| \geq N_{5.16}(\delta)(1+\varepsilon d/2)$, $|D_\mathbf{Q}| \geq N_{5.16}(\delta)$. Therefore there exists a σ-db $\mathscr{D}_\mathbf{P} \in \mathbf{P}$ such that $\mathrm{dist}_{+/-}(\mathscr{D}_\mathbf{P}, \mathscr{D}_\mathbf{Q}) \leq \delta d \min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\}$ (and therefore $\big||D_\mathbf{Q}| - |D_\mathbf{P}|\big| \leq \delta d \min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\}$). Since $\mathscr{D}$ is ε-far from $\mathbf{P}$, $\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}_\mathbf{P}) > \varepsilon d \min\{|D|, |D_\mathbf{P}|\}$ and so

$$\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}_\mathbf{Q}) > \varepsilon d \min\{|D|, |D_\mathbf{P}|\} - \delta d \min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\}.$$

By Lemma 5.18 (with $\mathscr{D}_1 = \mathscr{D}_\mathbf{Q}$, $\mathscr{D}_2 = \mathscr{D}_\mathbf{P}$ and $\mathscr{D}_3 = \mathscr{D}$),

$$\min\{|D|, |D_\mathbf{P}|\} \geq (1-\delta d)\min\{|D|, |D_\mathbf{Q}|\}.$$

Furthermore we can show that

$$\min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\} \leq (1+\varepsilon d/2)\min\{|D|, |D_\mathbf{Q}|\}.$$

To see this consider the case when $|D|$ is the smallest out of $|D|$, $|D_\mathbf{Q}|$ and $|D_\mathbf{P}|$, then

$$\min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\} \leq |D_\mathbf{Q}| \leq |D|(1+\varepsilon d/2) = (1+\varepsilon d/2)\min\{|D|, |D_\mathbf{Q}|\}.$$

If $|D_\mathbf{Q}|$ or $|D_\mathbf{P}|$ are the smallest then the inequality clearly holds. Therefore,

$$\begin{aligned}
\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}_\mathbf{Q}) &> \varepsilon d \min\{|D|, |D_\mathbf{P}|\} - \delta d \min\{|D_\mathbf{P}|, |D_\mathbf{Q}|\} \\
&\geq (\varepsilon d(1-\delta d) - \delta d(1+\frac{\varepsilon d}{2}))\min\{|D|, |D_\mathbf{Q}|\} \\
&\geq \left(\varepsilon d - \frac{\varepsilon}{6} - \frac{\varepsilon^2 d}{4}\right)\min\{|D|, |D_\mathbf{Q}|\} \\
&\geq \left(\varepsilon d - \frac{\varepsilon d}{6} - \frac{\varepsilon d}{4}\right)\min\{|D|, |D_\mathbf{Q}|\}
\end{aligned}$$

$$\geq \frac{\varepsilon d}{2} \min\{|D|, |D_{\mathbf{Q}}|\}$$

by the choice of δ. Hence $\mathscr{D}$ is $\varepsilon/2$-far from every $\mathscr{D}_{\mathbf{Q}} \in \mathbf{Q}_\delta$ and so $\mathscr{D}$ is $\varepsilon/2$-far from $\mathbf{Q}_\delta$. As $\pi_{\varepsilon/2}$ is an $\varepsilon/2$-tester for $\mathbf{Q}_\delta$ on $\mathbf{C}$, with probability at least $2/3$ the tester will reject. □

5.5 Constant time testability of hyperfinite hereditary properties

To the best of our knowledge, in the bounded degree graph model, it has not been shown explicitly that hyperfinite hereditary properties are uniformly (in n) testable in constant time. Benjamini, Schramm and Shapira in [17] prove that every monotone hyperfinite property is uniformly testable in constant time (in the bounded degree graph model). Their tester starts by testing for hyperfiniteness (which they show can be done in constant time). Newman and Sohler in [57] prove that every hyperfinite property is non-uniformly (in n) testable. We are interested in obtaining testers that are uniform in n *and* run in constant time. Furthermore, Hassidim et al. in [45] show that it is possible to approximate the distance to non-degenerate hereditary properties for hyperfinite graphs in constant time.

With methods similar to [17] and [27] it can be shown that every hereditary hyperfinite property is uniformly testable in constant time in the BDRD model.

Theorem 5.21. *Every hyperfinite hereditary property $\mathbf{P} \subseteq \mathbf{C}_d$ is uniformly testable on $\mathbf{C}_d$ in constant time in the* BDRD *model.*

By Lemma 5.3 and Theorem 5.21, we immediately get that every hyperfinite hereditary property is uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model.

Theorem 5.22. *Every hyperfinite hereditary property $\mathbf{P} \subseteq \mathbf{C}_d$ is uniformly testable on $\mathbf{C}_d$ in constant time in the* $\text{BDRD}_{+/-}$ *model.*

In this section, we will sketch a proof of Theorem 5.21 which follows closely to the proof given in [17]. We will then give an alternative proof of Theorem 5.22. In our alternative proof, we show that every hereditary hyperfinite property is close to having semilinear neighbourhood histograms and hence by Theorem 5.19 is uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model.

5.5.1 In the classical bounded degree model

First, we start with some definitions which are based on those used in [17].

Let $b \in \mathbb{N}$, recall that $\Psi(b,\sigma)$ is a maximal set of non-isomorphic connected σ-dbs of size at most b. From now on we let $\Psi(b) := \Psi(b,\sigma)$. For each $S \subseteq \Psi(b)$, let $\mathscr{D}(S)$ be the disjoint union of the σ-dbs in S. Let $\mathbf{P}$ be some hereditary property. Then let $\Phi_{\mathbf{P}}(S)$ be the smallest integer g such that the σ-db obtained by taking g disjoint copies of $\mathscr{D}(S)$ is not in $\mathbf{P}$. If no such integer exists (i.e. every σ-db that only contains connected components isomorphic to those in S is in $\mathbf{P}$) then $\Phi_{\mathbf{P}}(S) = \infty$.

Definition 5.23. *For a fixed hereditary property* $\mathbf{P}$ *and* $b \in \mathbb{N}$, *let* $\Pi^b_{\mathbf{P}} = \{S \subseteq \Psi(b) \mid \Phi_{\mathbf{P}}(S) < \infty\}$. *We then define the function* $\Phi_{\mathbf{P}} : \mathbb{N} \mapsto \mathbb{N}$ *as follows:*

$$\Phi_{\mathbf{P}}(b) = \begin{cases} 0 & \text{if } \Pi^b_{\mathbf{P}} = \emptyset \\ \max\limits_{S \in \Pi^b_{\mathbf{P}}} \Phi_{\mathbf{P}}(S) & \text{otherwise} \end{cases}$$

Some notes on the function $\Phi_{\mathbf{P}}$

First note that the function $\Phi_{\mathbf{P}}(b)$ is well defined as the set $\Psi(b)$ is finite. All hereditary properties can be characterised by a set of (possibly infinite) forbidden induced sub-databases (see, e.g. [10]). We claim that the value of $\Phi_{\mathbf{P}}(b)$ is related to the number and sizes of the connected components in the set of forbidden induced sub-databases of $\mathbf{P}$. We will first demonstrate this in the following example.

Example 5.24. Let $\mathbf{P}$ be the property containing all bounded degree chordal graphs (a graph is *chordal* if every cycle of length four or greater has a chord, where a *chord* of a cycle is an edge that is not in the edge set of the cycle but has endpoints in the cycle). A graph $\mathscr{G}$ is chordal if and only if $\mathscr{G}$ does not contain a cycle of length four or greater as an induced subgraph. Therefore the set of forbidden induced subgraphs of $\mathbf{P}$ is the set of cycles of length four or greater, i.e. every forbidden induced subgraph has one connected component. For any $b \in \mathbb{N}$ and $S \subseteq \Psi(b)$, if one of the graphs in S contains a cycle of length four or greater as an induced subgraph then $\Phi_{\mathbf{P}}(S) = 1$, otherwise $\Phi_{\mathbf{P}}(S) = \infty$. Hence if $b > 3$, then $\Phi_{\mathbf{P}}(b) = 1$, otherwise $\Phi_{\mathbf{P}}(b) = 0$.

In general, $\Phi_{\mathbf{P}}(b)$ is bounded above by the maximum number of connected components of a σ-db in the set of forbidden induced sub-databases of $\mathbf{P}$ whose connected components are all of size at most b. Furthermore, if every σ-db in the set of forbidden induced sub-databases has a component of size greater than b then $\Phi_{\mathbf{P}}(b) = 0$. We will summarise this in the following.

Observation 5.25. Let $\mathbf{P} \subseteq \mathbf{C}$ be a hereditary property and let $b \in \mathbb{N}$. Let $\mathbf{Q}$ be the set of forbidden induced sub-databases of $\mathbf{P}$ and let $\mathbf{Q}_b \subseteq \mathbf{Q}$ contain all the databases in $\mathbf{Q}$ whose

connected components have size at most b. Let m be the maximum number of connected components of the databases in $\mathbf{Q}_b$. Then

1. $\Phi_{\mathbf{P}}(b) \leq m$, and
2. if $\mathbf{Q}_b = \emptyset$, then $\Phi_{\mathbf{P}}(b) = 0$.

Proof. Let us start by proving 1. Let $S \subseteq \Psi(b)$ where $\Phi_{\mathbf{P}}(S) = g < \infty$ (note that if there exists no such S, then $\Phi_{\mathbf{P}}(b) = 0 \leq m$). We will show that $m \geq g$. Let $\mathscr{D}_g$ and $\mathscr{D}_{g-1}$ be the databases that consists of g and $g-1$ disjoint copies of $\mathscr{D}(S)$ respectively. By definition, $\mathscr{D}_g \notin \mathbf{P}$ and $\mathscr{D}_{g-1} \in \mathbf{P}$. Hence there exists $\mathscr{D} \in \mathbf{Q}_b$ such that $\mathscr{D}$ is an induced sub-database of $\mathscr{D}_g$ but $\mathscr{D}$ is not an induced sub-database of $\mathscr{D}_{g-1}$. We claim that the number of connected components in $\mathscr{D}$ is at least g. To see this let us assume that the number of connected components in $\mathscr{D}$ is at most $g-1$. Let $\mathscr{D}'$ be a database which is formed as follows: For each connected component in $\mathscr{D}$, pick a database $\mathscr{A} \in S$ such that the connected component is an induced sub-database of $\mathscr{A}$ (note at least one such $\mathscr{A}$ exists), then add a disjoint copy of $\mathscr{A}$ to $\mathscr{D}'$. Clearly $\mathscr{D}$ is an induced sub-database of $\mathscr{D}'$, and so $\mathscr{D}' \notin \mathbf{P}$. Furthermore, $\mathscr{D}'$ has at most $g-1$ disjoint copies of each $\mathscr{A} \in S$ and so is an induced sub-database of $\mathscr{D}_{g-1}$. However this then implies that $\mathscr{D}_{g-1} \notin \mathbf{P}$, which is a contradiction. Hence the number of connected components in $\mathscr{D}$ is at least g, and therefore $m \geq g$ as required.

Now let us prove 2. Let $\mathbf{Q}_b = \emptyset$ and let $S \subseteq \Psi(b)$. Then since every database in $\mathbf{Q}$ has a connected component of size greater than b, for any $g \in \mathbb{N}$ the database that consists of g disjoint copies of $\mathscr{D}(S)$ will not be in $\mathbf{P}$. Hence $\Phi_{\mathbf{P}}(S) = \infty$ and therefore $\Phi_{\mathbf{P}}(b) = 0$. □

Note that for many well known hereditary graph properties (for example chordal, perfect, acyclic and bipartite graphs) the maximum number of connected components in the set of forbidden induced subgraphs is 1.

Sketch of the proof of Theorem 5.21

Let $\varepsilon \in (0,1]$ and let $\mathbf{P}$ be a hyperfinite hereditary property on $\mathbf{C}_d$. Let $\varepsilon_0 := \varepsilon_0(\varepsilon)$ be a carefully chosen constant and let k be such that any database in $\mathbf{P}$ is (ε_0, k)-hyperfinite. Let us start by describing the ε-tester for $\mathbf{P}$ on $\mathbf{C}_d$. Let $\mathscr{D}$ be the input database. The tester starts by deciding correctly with high probability whether $\mathscr{D}$ is (ε_0, k)-hyperfinite or not $(\varepsilon/2, k)$-hyperfinite (ε_0 is chosen in such a way that $\mathscr{D}$ cannot be both (ε_0, k)-hyperfinite and not $(\varepsilon/2, k)$-hyperfinite). This can be done in constant time and with constant query complexity in the BDRD model (by extending methods in [17]). This is an ε-tester for the property of being (ε_0, k)-hyperfinite since if $\mathscr{D}$ is ε-far from being (ε_0, k)-hyperfinite it is not $(\varepsilon/2, k)$-hyperfinite. If $\mathscr{D}$ is declared to be not (ε_0, k)-hyperfinite then the tester rejects. If $\mathscr{D}$

is declared to be $(\varepsilon/2,k)$-hyperfinite then the tester samples a constant number $m = m(\varepsilon)$ of elements from $\mathscr{D}$ and for each element the tester explores its k-neighbourhood. If the induced sub-database of $\mathscr{D}$ on the union of the k-neighbourhoods of the sampled elements is not in $\mathbf{P}$, then the tester rejects. Otherwise, the tester accepts. This can be done with constant running time and constant query complexity.

Now to prove correctness (which follows closely to that in [17]), let us assume that $\mathscr{D} \in \mathbf{P}$. By the choice of k, $\mathscr{D}$ is (ε_0,k)-hyperfinite and so with high probability will be accepted in the first step of the tester. Then since $\mathbf{P}$ is hereditary the second step of the tester will accept with probability 1.

Let us now assume that $\mathscr{D}$ is ε-far from $\mathbf{P}$. Let us assume that $\mathscr{D}$ is $(\varepsilon/2,k)$-hyperfinite (otherwise the tester would reject with high probability). Let $\mathscr{D}'$ be the database that is formed from $\mathscr{D}$ by removing the minimum number of tuples required (at most $\varepsilon n/2$ tuples) such that every connected component in $\mathscr{D}'$ is of size at most k. Note that $\mathscr{D}'$ is $\varepsilon/2$-far from $\mathbf{P}$. Let $S \subseteq \Psi(k)$ be such that each $\mathscr{A} \in \Psi(k)$ is in S if and only if there are at least $\varepsilon n/4k|\Psi(k)|$ connected components in $\mathscr{D}'$ isomorphic to $\mathscr{A}$. We then let $\mathscr{D}''$ be the database formed from $\mathscr{D}'$ as follows. For every $\mathscr{A} \in \Psi(k)$, if $\mathscr{A} \notin S$, then remove every tuple in $\mathscr{D}'$ that is in a connected component isomorphic to $\mathscr{A}$. It is easy to see that the total number of tuples removed is at most $\varepsilon dn/4$ and therefore $\mathscr{D}''$ is $\varepsilon/4$-far from $\mathbf{P}$. Since $\mathscr{D}'' \notin \mathbf{P}$ and $\mathbf{P}$ is hereditary, $\Phi_{\mathbf{P}}(S) \leq \Phi_{\mathbf{P}}(k) < \infty$. Since each $\mathscr{A} \in S$ appears at least $\varepsilon n/4k|\Psi(k)|$ times in $\mathscr{D}''$ and $D'' = D$ we can choose m, the number of elements the tester samples, carefully to ensure that with high probability for each $\mathscr{A} \in S$ the tester samples at least $\Phi_{\mathbf{P}}(S)$ elements from connected components in $\mathscr{D}''$ that are isomorphic to $\mathscr{A}$. Furthermore, if we assume n is greater than some function of ε then with high probability each sampled element is from a distinct connected component (in $\mathscr{D}''$) and their $k+1$-neighbourhoods don't intersect (in $\mathscr{D}$). Let $a_1,\ldots a_m$ be the elements sampled in the tester. Let $\mathscr{D}_0$ be the induced sub-database of $\mathscr{D}$ on the union of the k-neighbourhoods of $a_1,\ldots a_m$ and let $\mathscr{D}_0''$ be the union of the connected components of $\mathscr{D}''$ that contain an element a_i. With high probability, by definition, $\mathscr{D}_0'' \notin \mathbf{P}$. Let $a \in D$. It is easy to see that the connected component in $\mathscr{D}''$ containing a is an induced sub-database of the k-neighbourhood of a in $\mathscr{D}$ (since $\mathscr{D}'$ was formed with the minimum required number of tuple deletions). Therefore, if none of the $k+1$-neighbourhoods of $a_1,\ldots a_m$ in $\mathscr{D}$ intersect (which happens with high probability), $\mathscr{D}_0''$ is an induced sub-database of $\mathscr{D}_0$. Finally since $\mathbf{P}$ is hereditary and with high probability $\mathscr{D}_0'' \notin \mathbf{P}$, with high probability $\mathscr{D}_0 \notin \mathbf{P}$ and the tester rejects.

5.5.2 In the new model

We will show that for every hyperfinite hereditary property $\mathbf{P}$ and $\delta \in (0,1]$ there exists a property $\mathbf{Q}$ that has a semilinear set of r-histograms and is δ-indistinguishable from $\mathbf{P}$.

Lemma 5.26. *Let $\mathbf{P} \subseteq \mathbf{C}_d$ be a hyperfinite hereditary property and let $\delta \in (0,1]$. Let ρ be the function such that $\mathbf{P}$ is ρ-hyperfinite on $\mathbf{C}_d$ and let*

$$b := \rho\Big(\frac{\delta d}{2(1+\delta d)}\Big).$$

Let $\mathbf{Q} \subseteq \mathbf{P}$ be the property such that for every $\mathscr{D} \in \mathbf{P}$, $\mathscr{D} \in \mathbf{Q}$ if and only if all connected components in $\mathscr{D}$ are of size at most b and for each $\mathscr{A} \in \Psi(b)$, $\mathscr{D}$ has either 0 or at least $\Phi_{\mathbf{P}}(b)$ connected components isomorphic to $\mathscr{A}$. Then

1. *$\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable, and*
2. *for every $r \in \mathbb{N}$, $\mathrm{h}_r(\mathbf{Q})$ is semilinear.*

Proof. Let us start by proving 1 of the lemma statement. Let

$$N := \frac{2(1+\delta d)\cdot \Phi_{\mathbf{P}}(b)\cdot |\Psi(b)|\cdot b}{\delta d}.$$

We will show that $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable with $N_{5.16}(\delta) = N$. If $\mathscr{D} \in \mathbf{Q}$ then $\mathscr{D} \in \mathbf{P}$ as $\mathbf{Q} \subseteq \mathbf{P}$. Hence to prove $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable, we only need to show that for every $\mathscr{D} \in \mathbf{P}$ with $|D| = n \geq N$ there exists a σ-db $\mathscr{D}' \in \mathbf{Q}$ such that $\mathrm{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \delta d \min\{n,|D'|\}$. Let $\mathscr{D} \in \mathbf{P}$ with $|D| = n > N$. Since $\mathbf{P}$ is ρ-hyperfinite on $\mathbf{C}$, by removing at most $\delta dn/2(1+\delta d)$ tuples from $\mathscr{D}$ we can obtain a σ-db $\mathscr{D}_1 \in \mathbf{C}$ that has connected components of size at most b (by the choice of b). For every tuple $\bar{a}$ that was removed from $\mathscr{D}$ to form $\mathscr{D}_1$, pick one element from $\bar{a}$ and remove it (and as a result any tuple containing that element) from $\mathscr{D}$. Let $\mathscr{D}_2$ be the resulting σ-db. Note that as $\mathbf{P}$ is hereditary, $\mathscr{D}_2 \in \mathbf{P}$. Furthermore, $\mathscr{D}_2$ has connected components of size at most b (as $\mathscr{D}_2$ is a sub-database of $\mathscr{D}_1$) and as at most $\delta dn/2(1+\delta d)$ elements were removed from $\mathscr{D}$ to form $\mathscr{D}_2$,

$$\mathrm{dist}_{+/-}(\mathscr{D},\mathscr{D}_2) \leq \frac{\delta dn}{2(1+\delta d)} \text{ and } |D_2| \geq \Big(1-\frac{\delta d}{2(1+\delta d)}\Big)n = \frac{n(2+\delta d)}{2(1+\delta d)}.$$

Now for every $\mathscr{A} \in \Psi(b)$, if $\mathscr{D}_2$ contains less than $\Phi_{\mathbf{P}}(b)$ many connected components isomorphic to $\mathscr{A}$, remove all such connected components from $\mathscr{D}_2$. Let $\mathscr{D}'$ be the resulting σ-db. Since $\mathbf{P}$ is hereditary, $\mathscr{D}' \in \mathbf{P}$ and by the construction of $\mathscr{D}'$, all connected components in $\mathscr{D}'$ are of size at most b and for each $\mathscr{A} \in \Psi(b)$, $\mathscr{D}'$ has either 0 or at least $\Phi_{\mathbf{P}}(b)$ connected

components isomorphic to $\mathscr{A}$. Therefore, $\mathscr{D}' \in \mathbf{Q}$. At most $\Phi_{\mathbf{P}}(b) \cdot |\Psi(b)|$ many connected components were removed from $\mathscr{D}_2$ to form $\mathscr{D}'$ and hence

$$|D'| \geq |D_2| - \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \geq \frac{n(2+\delta d)}{2(1+\delta d)} - \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b$$

and $\mathrm{dist}_{+/-}(\mathscr{D}_2, \mathscr{D}') \leq \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b$. Therefore,

$$\begin{aligned}
&\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \frac{\delta d n}{2(1+\delta d)} + \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \\
&\leq \frac{\delta d n}{2(1+\delta d)} + \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b + \frac{\delta d n}{2} - (1+\delta d) \cdot \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \\
&= \frac{\delta d n(2+\delta d)}{2(1+\delta d)} - \delta d \cdot \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \\
&\leq \delta d |D'|
\end{aligned}$$

as $n \geq N$ and so $\frac{\delta d n}{2} - (1+\delta d) \cdot \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \geq 0$. When constructing $\mathscr{D}'$ from $\mathscr{D}$ we only removed elements and hence $|D'| \leq |D|$ (and so $|D'| = \min\{|D|, |D'|\}$). Therefore $\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \delta d \min\{|D|, |D'|\}$. This completes the proof that $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable.

We will now prove 2 of the lemma statement. Let $r \in \mathbb{N}$ and for every $S \subseteq \Psi(b)$ let $\mathbf{Q}_S \subseteq \mathbf{Q}$ be the set of σ-dbs such that for every $\mathscr{D} \in \mathbf{Q}$, $\mathscr{D} \in \mathbf{Q}_S$ if and only if $\mathscr{D}$ contains at least $\Phi_{\mathbf{P}}(b)$ many connected components isomorphic to every $\mathscr{A} \in S$ but does not contain a connected component isomorphic to a σ-db in $\Psi(b) \setminus S$. Note that

$$\mathbf{Q} = \bigcup_{S \subseteq \Psi(b)} \mathbf{Q}_S.$$

We will prove that for every $S \subseteq \Psi(b)$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty and if $\Phi_{\mathbf{P}}(S) = \infty$, then $\mathrm{h}_r(\mathbf{Q}_S)$ is a linear set. Since $\mathbf{Q}$ is the union of the sets $\mathbf{Q}_S$, this will imply that $\mathrm{h}_r(\mathbf{Q})$ is a semilinear set.

Firstly let us prove that for every $S \subseteq \Psi(b)$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty. For a contradiction assume that for some $S \subseteq \Psi(b)$, $\Phi_{\mathbf{P}}(S) < \infty$ and there exists a σ-db $\mathscr{D} \in \mathbf{Q}_S$. Let $\mathscr{D}'$ be the σ-db that for each $\mathscr{A} \in S$ contains exactly $\Phi_{\mathbf{P}}(b)$ connected components isomorphic to $\mathscr{A}$ and contains no other connected components. By the definition of $\Phi_{\mathbf{P}}(b)$ and as $\Phi_{\mathbf{P}}(S) < \infty$, $\mathscr{D}' \notin \mathbf{P}$. However, $\mathscr{D}'$ is an induced sub-database of $\mathscr{D}$ and since $\mathbf{P}$ is hereditary and $\mathscr{D} \in \mathbf{P}$ (as $\mathbf{Q}_S \subseteq \mathbf{P}$) this implies $\mathscr{D}' \in \mathbf{P}$ which is a contradiction. Hence for every $S \subseteq \Psi(b)$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty.

We will now prove that for every $S \subseteq \Psi(b)$ if $\Phi_{\mathbf{P}}(S) = \infty$, then $\mathrm{h}_r(\mathbf{Q}_S)$ is a linear set. Let $S = \{\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_\ell\} \subseteq \Psi(b)$ be such that $\Phi_{\mathbf{P}}(S) = \infty$. Let $\bar{v} = \sum_{1 \leq i \leq \ell} \Phi_{\mathbf{P}}(b)\, \mathrm{h}_r(\mathscr{D}_i)$ and let

$$M = \{\bar{v} + a_1\, \mathrm{h}_r(\mathscr{D}_1) + a_2\, \mathrm{h}_r(\mathscr{D}_2) + \cdots + a_\ell\, \mathrm{h}_r(\mathscr{D}_\ell) \mid a_1, \ldots, a_\ell \in \mathbb{N}\}.$$

Clearly M is a linear set and we claim that $M = \mathrm{h}_r(\mathbf{Q}_S)$. Let $\bar{u} = \bar{v} + u_1\, \mathrm{h}_r(\mathscr{D}_1) + u_2\, \mathrm{h}_r(\mathscr{D}_2) + \cdots + u_\ell\, \mathrm{h}_r(\mathscr{D}_\ell) \in M$ for some $u_1, \ldots, u_\ell \in \mathbb{N}$ and let $\mathscr{D}$ be the σ-db with exactly $\Phi_{\mathbf{P}}(b) + u_1$ connected components isomorphic to $\mathscr{D}_1$, $\Phi_{\mathbf{P}}(b) + u_2$ connected components isomorphic to $\mathscr{D}_2$, …, $\Phi_{\mathbf{P}}(b) + u_\ell$ connected components isomorphic to $\mathscr{D}_\ell$ and no other connected components. Clearly $\bar{u} = \mathrm{h}_r(\mathscr{D})$. Then as P is hereditary and $\Phi_{\mathbf{P}}(S) = \infty$, $\mathscr{D} \in \mathbf{P}$ and so by the definition of $\mathbf{Q}_S$, $\mathscr{D} \in \mathbf{Q}_S$. On the other hand let $\mathscr{D} \in \mathbf{Q}_S$ then by definition for some $u_1, \ldots, u_\ell \in \mathbb{N}$, $\mathscr{D}$ contains exactly $\Phi_{\mathbf{P}}(b) + u_1$ connected components isomorphic to $\mathscr{D}_1$, $\Phi_{\mathbf{P}}(b) + u_2$ connected components isomorphic to $\mathscr{D}_2$, …, $\Phi_{\mathbf{P}}(b) + u_\ell$ connected components isomorphic to $\mathscr{D}_\ell$ and no other connected components. The r-histogram vector of $\mathscr{D}$ is then $\bar{v} + u_1\, \mathrm{h}_r(\mathscr{D}_1) + u_2\, \mathrm{h}_r(\mathscr{D}_2) + \cdots + u_\ell\, \mathrm{h}_r(\mathscr{D}_\ell)$ and hence $\mathrm{h}_r(\mathscr{D}) \in M$. Therefore $M = \mathrm{h}_r(\mathbf{Q}_S)$.

We have proven that for every $S \subseteq \Psi(b)$, $\mathrm{h}_r(\mathbf{Q}_S)$ is either empty or a linear set and hence $\mathrm{h}_r(\mathbf{Q})$ is semilinear. □

Combining Theorem 5.19 and Lemma 5.26 we obtain Theorem 5.22 as a corollary.

5.6 Using our techniques in the classical bounded degree model

One natural question is can any of the methods we have used in the $\mathrm{BDRD}_{+/-}$ model be used to also obtain constant time testers in the BDRD model. In this section, we will show that any hereditary and local (in the BDRD model) property whose r-histograms are close to being semilinear is constant time uniformly testable in the BDRD model (Theorem 5.32). To prove Theorem 5.32 we start by showing that for any property $\mathbf{P}$ which is hereditary and local (in the BDRD model) on the input class $\mathbf{C}$, if the r-histograms of $\mathbf{P}$ are close to being semilinear, then for every σ-db $\mathscr{D}$ in $\mathbf{P}$ there exists a constant size σ-db in $\mathbf{P}$ with a neighbourhood distribution similar to that of $\mathscr{D}$, but this is not true for σ-dbs in $\mathbf{C}$ that are far from $\mathbf{P}$ (Theorem 5.29). To prove Theorem 5.29 we use similar techniques to those used to prove Theorem 5.11. Let us start by defining indistinguishability in the BDRD model.

Definition 5.27 (δ-indistinguishable in the BDRD model)**.** *Let $\delta \in (0,1]$. Two properties $\mathbf{P}$ and $\mathbf{Q}$ are called* δ-indistinguishable in the BDRD model *if there exists $N := N_{5.27}(\delta) \in \mathbb{N}$ that satisfies the following. For every σ-db $\mathscr{D} \in \mathbf{P}$ with $|D| = n \geq N$ elements there exists*

a σ-db $\mathscr{D}' \in \mathbf{Q}$ such that $\operatorname{dist}(\mathscr{D},\mathscr{D}') \leq \delta dn$*; and for every σ-db $\mathscr{D} \in \mathbf{Q}$ with $|D| = n \geq N$ elements there exists a σ-db $\mathscr{D}' \in \mathbf{P}$ such that* $\operatorname{dist}(\mathscr{D},\mathscr{D}') \leq \delta dn$.

The following lemma will be useful in the proof of Theorem 5.29.

Lemma 5.28. *Let $\mathscr{D}$ and $\mathscr{D}'$ be two σ-dbs on n elements and let $\delta \in (0,1]$. If* $\operatorname{dist}(\mathscr{D},\mathscr{D}') \leq \delta dn$ *then for any $r \in \mathbb{N}$,*

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$$

and

$$\|\operatorname{h}_r(\mathscr{D}) - \operatorname{h}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n.$$

Proof. Let $\delta \in (0,1]$ and let $r \in \mathbb{N}$. Let us assume that $\operatorname{dist}(\mathscr{D},\mathscr{D}') \leq \delta dn$. The distance between $\mathscr{D}$ and $\mathscr{D}'$ is the minimum number of modifications needed to make $\mathscr{D}$ and $\mathscr{D}'$ isomorphic. The two different types of modifications allowed are: (1) inserting a new tuple, and (2) deleting a tuple. If a tuple $\bar{a}$ is inserted or deleted from $\mathscr{D}$ or $\mathscr{D}'$ then the r-type of any element that is at distance at most r from every element in $\bar{a}$ could have changed. Hence, since the number of elements in the r-neighbourhood of an element is at most $(\operatorname{ar}(\sigma)\cdot d)^{r+1}$ (by Lemma 2.16), every modification to $\mathscr{D}$ or $\mathscr{D}'$ could change the r-type of at most $(\operatorname{ar}(\sigma)\cdot d)^{r+1}$ many elements. Therefore, $\|\operatorname{h}_r(\mathscr{D}) - \operatorname{h}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n$ as required.

By definition,

$$\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 = \sum_{i=1}^{c(r)} \left|\frac{\operatorname{h}_r(\mathscr{D})[i]}{n} - \frac{\operatorname{h}_r(\mathscr{D}')[i]}{n}\right| = \frac{1}{n}\|\operatorname{h}_r(\mathscr{D}) - \operatorname{h}_r(\mathscr{D}')\|_1.$$

Hence, $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$ as required. □

Theorem 5.29. *Let $\mathbf{P} \subseteq \mathbf{C}$ be a hereditary property that is local on $\mathbf{C}$ in the* BDRD *model. Let us assume that for every $\delta \in (0,1]$ there exists a property $\mathbf{Q}_\delta \subseteq \mathbf{P}$ such that $\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable (in the* BDRD *model), and for every $r \in \mathbb{N}$, $\operatorname{h}_r(\mathbf{Q}_\delta)$ is semilinear. Then for every $\varepsilon \in (0,1]$ there exists $n_{min} := n_{min}(\varepsilon), n_{max} := n_{max}(\varepsilon) \in \mathbb{N}$ and $f := f(\varepsilon), \mu := \mu(\varepsilon) \in (0,1)$ such that for every $\mathscr{D} \in \mathbf{C}$ with $|D| > n_{max}$,*

1. *if $\mathscr{D} \in \mathbf{P}$, then there exists a $\mathscr{D}' \in \mathbf{P}$ such that $n_{min} \leq |D'| \leq n_{max}$ and $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq f - \mu$, and*
2. *if $\mathscr{D}$ is ε-far from $\mathbf{P}$ (in the* BDRD *model), then for every $\mathscr{D}' \in \mathbf{P}$ such that $n_{min} \leq |D'| \leq n_{max}$, we have $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 > f + \mu$.*

Proof. Let $\varepsilon \in (0,1]$. Let $N := N_{5.6}(\varepsilon)$, $\lambda := \lambda_{5.6}(\varepsilon)$ and $r := r_{5.6}(\varepsilon)$ be as in Definition 5.6 for $\mathbf{P}$ and ε. Let $c := c(r)$ and let

$$\delta := \frac{\lambda}{90cd^{r+2}\operatorname{ar}(\sigma)^{r+1}}.$$

Let us assume there exists a property $\mathbf{Q}_\delta \subseteq \mathbf{P}$ such that $\mathbf{P}$ is δ-indistinguishable from $\mathbf{Q}_\delta$ and $\mathrm{h}_r(\mathbf{Q}_\delta)$ is semilinear. Let $N_{\text{ind}} := N_{5.27}(\delta)$ be as in Definition 5.27 for $\mathbf{P}$, $\mathbf{Q}_\delta$ and δ.

First note that if $\mathbf{P}$ is empty then for any choice of $n_{\min}$, $n_{\max}$, f and μ, both 1 and 2 in the theorem statement are true and hence we shall assume that $\mathbf{P}$ (and hence $\mathbf{Q}_\delta$) is non-empty. As $\mathrm{h}_r(\mathbf{Q}_\delta)$ is a semilinear set we can write it as follows, $\mathrm{h}_r(\mathbf{Q}_\delta) = M_1 \cup M_2 \cup \cdots \cup M_m$ where $m \in \mathbb{N}$ and for each $i \in [m]$, $M_i = \{\vec{v}_0^i + a_1\vec{v}_1^i + \cdots + a_{k_i}\vec{v}_{k_i}^i \mid a_1, \ldots, a_{k_i} \in \mathbb{N}\}$ is a linear set where $\vec{v}_0^i, \ldots, \vec{v}_{k_i}^i \in \mathbb{N}^c$ and for each $j \in [k_i]$, $\|\vec{v}_j^i\|_1 \neq 0$. Let $k := \max_{i\in[m]} k_i + 1$ and $v := \max_{i\in[m]}\left(\max_{j\in[0,k_i]} \|\vec{v}_j^i\|_1\right)$ (note that $v > 0$ as $\mathbf{Q}_\delta$ is non-empty). Let

- $n_{\min} := n_0 - kv$,
- $n_{\max} := n_0 + kv$,
- $f := \frac{\lambda}{20c}$, and
- $\mu := \frac{\lambda}{40c}$

where

$$n_0 := \max\left\{N_{\text{ind}} + N + kv, kv\left(\frac{3ckv(\operatorname{ar}(\sigma)\cdot d)^{r+1}}{f - \mu - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}} + 1\right)\right\}.$$

Note that $n_{\min} > 0$ by the choice of n_0, f, μ and δ.

We will start by proving 1 of the lemma statement. Let $\mathscr{D} \in \mathbf{P}$ with $|D| = n > n_{\max}$. Then since $\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable and $n_{\max} \geq N_{\text{ind}}$, there exists $\mathscr{D}_1 \in \mathbf{Q}_\delta$ such that $\operatorname{dist}(\mathscr{D}, \mathscr{D}_1) \leq \delta dn$. By Lemma 5.28, $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}_1)\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$. Since $\mathrm{h}_r(\mathbf{Q}_\delta)$ is semilinear and by the choices of n_0, $n_{\min}$ and $n_{\max}$, by Theorem 4.2 there exists a σ-db $\mathscr{D}' \in \mathbf{Q}_\delta$ such that $n_{\min} \leq |D'| \leq n_{\max}$ and $\|\operatorname{dv}_r(\mathscr{D}_1) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq f - \mu - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$. Therefore, $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq f - \mu$ as required.

Next let us prove 2 of the lemma statement. Let $\mathscr{D} \in \mathbf{C}$ be ε-far from $\mathbf{P}$ with $|D| = n > n_{\max}$. For a contradiction let us assume there does exist a σ-db $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$ and $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq f + \mu$. Then since $\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable and $|D'| \geq n_{\min} \geq N_{\text{ind}}$, there exists a σ-db $\mathscr{D}'_{\mathbf{Q}} \in \mathbf{Q}_\delta$ such that $|D'_{\mathbf{Q}}| = |D'|$ and $\operatorname{dist}(\mathscr{D}'_{\mathbf{Q}}, \mathscr{D}') \leq \delta d|D'|$. By Lemma 5.28, $\|\operatorname{dv}_r(\mathscr{D}'_{\mathbf{Q}}) - \operatorname{dv}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}$. Since $\mathscr{D}'_{\mathbf{Q}} \in \mathbf{Q}_\delta$

there exists some $i \in [m]$ and $a_1^{\mathscr{D}'\mathbf{Q}}, \ldots, a_{k_i}^{\mathscr{D}'\mathbf{Q}} \in \mathbb{N}$ such that

$$\mathrm{h}_r(\mathscr{D}'_{\mathbf{Q}}) = \vec{v}_0^i + a_1^{\mathscr{D}'\mathbf{Q}}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}'\mathbf{Q}}\vec{v}_{k_i}^i.$$

Let $\mathscr{D}_{\mathbf{Q}}$ be the σ-db with r-histogram

$$\mathrm{h}_r(\mathscr{D}_{\mathbf{Q}}) = \vec{v}_0^i + a_1^{\mathscr{D}_{\mathbf{Q}}}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}_{\mathbf{Q}}}\vec{v}_{k_i}^i \in M_i$$

where $a_j^{\mathscr{D}_{\mathbf{Q}}}$ is the nearest integer to

$$\Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big) \cdot \frac{n a_j^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|}.$$

Note as $\vec{v}_0^i + a_1^{\mathscr{D}_{\mathbf{Q}}}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}_{\mathbf{Q}}}\vec{v}_{k_i}^i \in \mathrm{h}_r(\mathbf{Q}_\delta)$, $\mathscr{D}_{\mathbf{Q}}$ exists and $\mathscr{D}_{\mathbf{Q}} \in \mathbf{Q}_\delta$. We claim that $n \leq |D_{\mathbf{Q}}| \leq kv + (1+\lambda/6c(\mathrm{ar}(\sigma)\cdot d)^{r+1})n$ and $\|\mathrm{h}_r(\mathscr{D}_{\mathbf{Q}}) - \mathrm{h}_r(\mathscr{D})\|_1 \leq \frac{\lambda n}{3}$.

Claim 5.30. $n \leq |D_{\mathbf{Q}}| \leq kv + (1+\lambda/6c(\mathrm{ar}(\sigma)\cdot d)^{r+1})n$.

Proof: First note that $|D_{\mathbf{Q}}| = \|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}_{\mathbf{Q}}}\|\vec{v}_\ell^i\|_1$. By the choice of $a_\ell^{\mathscr{D}_{\mathbf{Q}}}$ for $\ell \in [k_i]$,

$$\begin{aligned}
&\|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}_{\mathbf{Q}}}\|\vec{v}_\ell^i\|_1 \\
&\leq \|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} \Big(\Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{n a_\ell^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|} + \frac{1}{2}\Big)\|\vec{v}_\ell^i\|_1 \\
&= \|\vec{v}_0^i\|_1 + \frac{1}{2}\sum_{\ell\in[k_i]}\|\vec{v}_\ell^i\|_1 + \Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{n}{|D'_{\mathbf{Q}}|}\sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'\mathbf{Q}}\|\vec{v}_\ell^i\|_1 \\
&\leq kv + \Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)n
\end{aligned}$$

Similarly, by the choice of $a_\ell^{\mathscr{D}_{\mathbf{Q}}}$ for $\ell \in [k_i]$,

$$\begin{aligned}
&\|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}_{\mathbf{Q}}}\|\vec{v}_\ell^i\|_1 \\
&\geq \|\vec{v}_0^i\|_1 + \sum_{\ell\in[k_i]} \Big(\Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{n a_\ell^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|} - \frac{1}{2}\Big)\|\vec{v}_\ell^i\|_1 \\
&= \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{\ell\in[k_i]}\|\vec{v}_\ell^i\|_1 + \Big(1 + \frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{n}{|D'_{\mathbf{Q}}|}\sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'\mathbf{Q}}\|\vec{v}_\ell^i\|_1
\end{aligned}$$

$$= \|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{\ell\in[k_i]}\|\vec{v}_\ell^i\|_1 + n\Big(1+\frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\Big(1-\frac{\|\vec{v}_0^i\|_1}{|D'_{\mathbf{Q}}|}\Big)$$

since $\sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'\mathbf{Q}}\|\vec{v}_\ell^i\|_1 = |D'_{\mathbf{Q}}| - \|\vec{v}_0^i\|_1$. Then since $|D'_{\mathbf{Q}}| \geq n_{\min} \geq 12\|\vec{v}_0^i\|_1 c(\mathrm{ar}(\sigma)\cdot d)^{r+1}/\lambda$,

$$\begin{aligned}
&\|\vec{v}_0^i\|_1 - \frac{1}{2}\sum_{\ell\in[k_i]}\|\vec{v}_\ell^i\|_1 + n\Big(1+\frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\Big(1-\frac{\|\vec{v}_0^i\|_1}{|D'_{\mathbf{Q}}|}\Big)\\
&\geq -kv + n\Big(1+\frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\Big(1-\frac{\lambda}{12c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\\
&= -kv + n + n\Big(\frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}} - \frac{\lambda}{12c(\mathrm{ar}(\sigma)\cdot d)^{r+1}} - \frac{\lambda^2}{72c^2(\mathrm{ar}(\sigma)\cdot d)^{2r+2}}\Big)\\
&\geq -kv + n + \frac{\lambda n}{24c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\\
&\geq -kv + n + kv = n
\end{aligned}$$

as $n > n_{\max} \geq 24ckv(\mathrm{ar}(\sigma)\cdot d)^{r+1}/\lambda$. ■

Claim 5.31. $\|\mathrm{h}_r(\mathscr{D}_{\mathbf{Q}}) - \mathrm{h}_r(\mathscr{D})\|_1 \leq \frac{\lambda n}{3}$.

Proof: First note that as $\|\mathrm{dv}_r(\mathscr{D}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq f+\mu$ and $\|\mathrm{dv}_r(\mathscr{D}'_{\mathbf{Q}}) - \mathrm{dv}_r(\mathscr{D}')\|_1 \leq 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}$ we have $\|\mathrm{dv}_r(\mathscr{D}'_{\mathbf{Q}}) - \mathrm{dv}_r(\mathscr{D})\|_1 \leq 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1} + f + \mu$. Then since $\mathrm{h}_r(\mathscr{D}'_{\mathbf{Q}}) = \vec{v}_0^i + a_1^{\mathscr{D}'\mathbf{Q}}\vec{v}_1^i + \cdots + a_{k_i}^{\mathscr{D}'\mathbf{Q}}\vec{v}_{k_i}^i$, for every $j\in[c]$

$$\mathrm{h}_r(\mathscr{D})[j] \geq n\Big(\frac{\vec{v}_0^i[j] + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'\mathbf{Q}}\vec{v}_\ell^i[j]}{|D'_{\mathbf{Q}}|} - f - \mu - 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\Big)$$

and

$$\mathrm{h}_r(\mathscr{D})[j] \leq n\Big(\frac{\vec{v}_0^i[j] + \sum_{\ell\in[k_i]} a_\ell^{\mathscr{D}'\mathbf{Q}}\vec{v}_\ell^i[j]}{|D'_{\mathbf{Q}}|} + f + \mu + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}\Big).$$

Hence, by the choice of $a_\ell^{\mathscr{D}\mathbf{Q}}$ for $\ell\in[k_i]$,

$$\begin{aligned}
&\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}_{\mathbf{Q}})[j]\\
&\leq \vec{v}_0^i[j]\Big(\frac{n}{|D'_{\mathbf{Q}}|} - 1\Big) + \sum_{\ell\in[k_i]}\vec{v}_\ell^i[j]\Big(\frac{na_\ell^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|} - a_\ell^{\mathscr{D}\mathbf{Q}}\Big) + fn + \mu n + 2\delta d^{r+2}\,\mathrm{ar}(\sigma)^{r+1}n\\
&\leq \frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} + \sum_{\ell\in[k_i]}\vec{v}_\ell^i[j]\Big(\frac{na_\ell^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|} - \Big(\Big(1+\frac{\lambda}{6c(\mathrm{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{na_\ell^{\mathscr{D}'\mathbf{Q}}}{|D'_{\mathbf{Q}}|} - \frac{1}{2}\Big)\Big)
\end{aligned}$$

$$
\begin{aligned}
&+ fn + \mu n + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&\leq \frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} + \frac{1}{2}\sum_{\ell\in[k_i]}\vec{v}_\ell^i[j] + \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} + fn + \mu n + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n.
\end{aligned}
$$

Similarly, by the choice of $a_\ell^{\mathscr{D}_{\mathbf{Q}}}$ for $\ell \in [k_i]$ and as $n > |D'_{\mathbf{Q}}|$,

$$
\begin{aligned}
&\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}_{\mathbf{Q}})[j] \\
&\geq \vec{v}_0^i[j]\Big(\frac{n}{|D'_{\mathbf{Q}}|} - 1\Big) + \sum_{\ell\in[k_i]}\vec{v}_\ell^i[j]\Big(\frac{na_\ell^{\mathscr{D}'_{\mathbf{Q}}}}{|D'_{\mathbf{Q}}|} - a_\ell^{\mathscr{D}_{\mathbf{Q}}}\Big) - fn - \mu n - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&\geq -\frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} + \sum_{\ell\in[k_i]}\vec{v}_\ell^i[j]\Big(\frac{na_\ell^{\mathscr{D}'_{\mathbf{Q}}}}{|D'_{\mathbf{Q}}|} - \Big(\Big(1 + \frac{\lambda}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}}\Big)\cdot\frac{na_\ell^{\mathscr{D}'_{\mathbf{Q}}}}{|D'_{\mathbf{Q}}|} + \frac{1}{2}\Big)\Big) \\
&- fn - \mu n - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&= -\frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} - \frac{1}{2}\sum_{\ell\in[k_i]}\vec{v}_\ell^i[j] - \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}|D'_{\mathbf{Q}}|}\sum_{\ell\in[k_i]}\vec{v}_\ell^i[j]a_\ell^{\mathscr{D}'_{\mathbf{Q}}} \\
&- fn - \mu n - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&\geq -\frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} - \frac{1}{2}\sum_{\ell\in[k_i]}\vec{v}_\ell^i[j] - \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} - fn - \mu n - 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&|\mathrm{h}_r(\mathscr{D})[j] - \mathrm{h}_r(\mathscr{D}_{\mathbf{Q}})[j]| \\
&\leq \frac{n\vec{v}_0^i[j]}{|D'_{\mathbf{Q}}|} + \frac{1}{2}\sum_{\ell\in[k_i]}\vec{v}_\ell^i[j] + \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} + fn + \mu n + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&\leq \frac{n}{|D'_{\mathbf{Q}}|}\sum_{0\leq\ell\leq k_i}\vec{v}_\ell^i[j] + \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} + fn + \mu n + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&\leq \frac{nkv}{|D'_{\mathbf{Q}}|} + \frac{\lambda n}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} + fn + \mu n + 2\delta d^{r+2}\operatorname{ar}(\sigma)^{r+1}n \\
&= n\Big(\frac{kv}{|D'_{\mathbf{Q}}|} + \frac{\lambda}{6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}} + \frac{\lambda}{20c} + \frac{\lambda}{40c} + \frac{\lambda}{45c}\Big) \\
&\leq n\Big(\frac{\lambda}{20c} + \frac{\lambda}{6c} + \frac{\lambda}{20c} + \frac{\lambda}{40c} + \frac{\lambda}{45c}\Big) \\
&\leq \frac{\lambda n}{3c}
\end{aligned}
$$

by the choice of f, μ and δ and as

$$|D'_{\mathbf{Q}}| \geq n_{\min} \geq \frac{20ckv}{\lambda}.$$

Hence,

$$\|\operatorname{h}_r(\mathscr{D}_{\mathbf{Q}}) - \operatorname{h}_r(\mathscr{D})\|_1 \leq \frac{\lambda n}{3}$$

as required. ■

Let $\mathscr{D}''$ be the σ-db on n elements formed from $\mathscr{D}_{\mathbf{Q}}$ by removing $|D_{\mathbf{Q}}| - n$ many elements at random (note that by Claim 5.30, $|D_{\mathbf{Q}}| \geq n$). Since $\mathscr{D}_{\mathbf{Q}} \in \mathbf{Q}_\delta \subseteq \mathbf{P}$ and $\mathbf{P}$ is hereditary, $\mathscr{D}'' \in \mathbf{P}$. Furthermore, since $|D_{\mathbf{Q}}| - n \leq kv + \lambda n/6c(\operatorname{ar}(\sigma)\cdot d)^{r+1}$ by Claim 5.30, and each element that was removed from $\mathscr{D}_{\mathbf{Q}}$ potentially causes the r-type of at most $(\operatorname{ar}(\sigma)\cdot d)^{r+1}$ elements to change (by Lemma 2.16), $\|\operatorname{h}_r(\mathscr{D}_{\mathbf{Q}}) - \operatorname{h}_r(\mathscr{D}'')\|_1 \leq 2kv(\operatorname{ar}(\sigma)\cdot d)^{r+1} + \lambda n/3c$. Therefore by Claim 5.31,

$$\|\operatorname{h}_r(\mathscr{D}) - \operatorname{h}_r(\mathscr{D}'')\|_1 \leq 2kv(\operatorname{ar}(\sigma)\cdot d)^{r+1} + \frac{\lambda n}{3c} + \frac{\lambda n}{3} \leq \lambda n$$

since $n > n_{\max} \geq 6kv(\operatorname{ar}(\sigma)\cdot d)^{r+1}/\lambda$. Finally, since $\mathbf{P}$ is local on $\mathbf{C}$ in the BDRD model, $|D''| = n > n_{\max} \geq N$ and $\mathscr{D}'' \in \mathbf{P}$, we get that $\mathscr{D}$ is ε-close to $\mathbf{P}$ in the BDRD model which is a contradiction. Therefore for every $\mathscr{D}' \in \mathbf{P}$ such that $n_{\min} \leq |D'| \leq n_{\max}$, we have $\|\operatorname{dv}_r(\mathscr{D}) - \operatorname{dv}_r(\mathscr{D}')\|_1 > f + \mu$. □

Using Theorem 5.29, we can now show that any hereditary and local (in the BDRD model) property whose r-histograms are close to being semilinear is constant time uniformly testable in the BDRD model. The proof of Theorem 5.32 is very similar to the proof of Theorem 5.13.

Theorem 5.32. *Let $\mathbf{P} \subseteq \mathbf{C}$ be a hereditary property that is local on $\mathbf{C}$ in the* BDRD *model. If for every $\delta \in (0,1]$ there exists a property $\mathbf{Q}_\delta \subseteq \mathbf{P}$ such that*

1. *$\mathbf{P}$ and $\mathbf{Q}_\delta$ are δ-indistinguishable (in the* BDRD *model), and*
2. *for every $r \in \mathbb{N}$, $\operatorname{h}_r(\mathbf{Q}_\delta)$ is semilinear*

then $\mathbf{P}$ is uniformly testable on $\mathbf{C}$ in constant time in the BDRD *model.*

Proof. Let $\varepsilon \in (0,1]$. Then using very similar arguments as in the proof of Theorem 5.13, but using Theorem 5.29 instead of Theorem 5.11, there exists an ε-tester for $\mathbf{P}$ on $\mathbf{C}$ (in the BDRD model) that has constant query complexity and constant running time. □

5.6.1 Monotone hyperfinite properties

As discussed earlier, in [17] a proof for the constant time testability of monotone hyperfinite properties in the bounded degree graph model was given. We extended this proof to the BDRD model (and to hyperfinite hereditary properties) in Theorem 5.21. We will give an alternative proof of the constant time testability of monotone hyperfinite properties in the BDRD model. We start by showing that for every monotone hyperfinite property $\mathbf{P}$ and $\delta \in (0,1]$ there exists a sub-property of $\mathbf{P}$ that has a semilinear set of r-histograms and is δ-indistinguishable from $\mathbf{P}$ in the BDRD model (Lemma 5.33).

Lemma 5.33. *Let $\mathbf{P} \subseteq \mathbf{C}$ be a monotone property which is hyperfinite on $\mathbf{C}$ and let $\delta \in (0,1]$. Let ρ be the function such that $\mathbf{P}$ is ρ-hyperfinite on $\mathbf{C}$ and let*

$$b := \rho\left(\frac{\delta}{2}\right).$$

Let $\mathbf{Q} \subseteq \mathbf{P}$ be defined as follows: For every $\mathscr{D} \in \mathbf{P}$, $\mathscr{D} \in \mathbf{Q}$ if and only if all connected components in $\mathscr{D}$ are of size at most b and for each $\mathscr{A} \in \Psi(b) \setminus \{\mathscr{B}\}$, where $\mathscr{B} \in \Psi(b)$ is the σ-db with exactly one element and no tuples, $\mathscr{D}$ has either 0 or at least $\Phi_{\mathbf{P}}(b)$ connected components isomorphic to $\mathscr{A}$. Then

1. *$\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable in the* BDRD *model, and*
2. *for every $r \in \mathbb{N}$, $\mathrm{h}_r(\mathbf{Q})$ is semilinear.*

Proof. Let us start by proving 1 of the lemma statement. Let

$$N := \frac{2 \cdot \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b}{\delta}.$$

We will show that $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable in the BDRD model with $N_{5.27}(\delta) = N$. If $\mathscr{D} \in \mathbf{Q}$ then $\mathscr{D} \in \mathbf{P}$ as $\mathbf{Q} \subseteq \mathbf{P}$. Hence to prove $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable, we only need to show that for every $\mathscr{D} \in \mathbf{P}$ with $|D| = n \geq N$ there exists a σ-db $\mathscr{D}' \in \mathbf{Q}$ such that $\mathrm{dist}(\mathscr{D}, \mathscr{D}') \leq \delta dn$. Let $\mathscr{D} \in \mathbf{P}$ with $|D| = n > N$. Since $\mathbf{P}$ is ρ-hyperfinite on $\mathbf{C}$, by removing at most $\delta n/2$ tuples from $\mathscr{D}$ we can obtain a σ-db $\mathscr{D}_1 \in \mathbf{C}$ that has connected components of size at most b (by the choice of b). Let $\mathscr{B} \in \Psi(b)$ be the σ-db with exactly one element and no tuples. Now for every $\mathscr{A} \in \Psi(b) \setminus \{\mathscr{B}\}$, if $\mathscr{D}_1$ contains less than $\Phi_{\mathbf{P}}(b)$ many connected components isomorphic to $\mathscr{A}$, remove all tuples from such connected components from $\mathscr{D}_1$. Let $\mathscr{D}'$ be the resulting σ-db. Since $\mathbf{P}$ is monotone, $\mathscr{D}' \in \mathbf{P}$ and by the construction of $\mathscr{D}'$, all connected components in $\mathscr{D}'$ are of size at most b and for each $\mathscr{A} \in \Psi(b) \setminus \{\mathscr{B}\}$, $\mathscr{D}'$ has either 0 or at least $\Phi_{\mathbf{P}}(b)$ connected components isomorphic to $\mathscr{A}$. Therefore, $\mathscr{D}' \in \mathbf{Q}$.

At most $\Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \cdot d$ many tuples were removed from $\mathscr{D}_1$ to form $\mathscr{D}'$ and hence $\text{dist}(\mathscr{D}_1, \mathscr{D}') \leq \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \cdot d$. Therefore,

$$\begin{aligned}\text{dist}(\mathscr{D}, \mathscr{D}') &\leq \frac{\delta n}{2} + \Phi_{\mathbf{P}}(b) \cdot |\Psi(b)| \cdot b \cdot d \\ &\leq \frac{\delta nd}{2} + \frac{\delta nd}{2} \\ &= \delta dn\end{aligned}$$

since $n \geq N$ and by the choice of N. This completes the proof that $\mathbf{P}$ and $\mathbf{Q}$ are δ-indistinguishable in the BDRD model.

We will now prove 2 of the lemma statement. Let $r \in \mathbb{N}$. For every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, let $\mathbf{Q}_S \subseteq \mathbf{Q}$ be the set of σ-dbs such that for every $\mathscr{D} \in \mathbf{Q}$, $\mathscr{D} \in \mathbf{Q}_S$ if and only if $\mathscr{D}$ contains at least $\Phi_{\mathbf{P}}(b)$ many connected components isomorphic to every $\mathscr{A} \in S$ but does not contain a connected component isomorphic to a σ-db in $\Psi(b) \setminus (S \cup \{\mathscr{B}\})$. Note that

$$\mathbf{Q} = \bigcup_{S \subseteq \Psi(b) \setminus \{\mathscr{B}\}} \mathbf{Q}_S.$$

We will prove that for every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty and if $\Phi_{\mathbf{P}}(S) = \infty$, then $\text{h}_r(\mathbf{Q}_S)$ is a linear or semilinear set. Since $\mathbf{Q}$ is the union of the sets $\mathbf{Q}_S$, this will imply that $\text{h}_r(\mathbf{Q})$ is a semilinear set.

Firstly let us prove that for every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty. For a contradiction assume that for some $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, $\Phi_{\mathbf{P}}(S) < \infty$ and there exists a σ-db $\mathscr{D} \in \mathbf{Q}_S$. Let $\mathscr{D}'$ be the σ-db that for each $\mathscr{A} \in S$ contains exactly $\Phi_{\mathbf{P}}(b)$ connected components isomorphic to $\mathscr{A}$ and contains no other connected components. By the definition of $\Phi_{\mathbf{P}}(b)$ and as $\Phi_{\mathbf{P}}(S) < \infty$, $\mathscr{D}' \notin \mathbf{P}$. However, $\mathscr{D}'$ is a sub-database of $\mathscr{D}$ and since $\mathbf{P}$ is monotone and $\mathscr{D} \in \mathbf{P}$ (as $\mathbf{Q}_S \subseteq \mathbf{P}$) this implies $\mathscr{D}' \in \mathbf{P}$ which is a contradiction. Hence for every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, if $\Phi_{\mathbf{P}}(S) < \infty$, then $\mathbf{Q}_S$ is empty.

We will now prove that for every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$ if $\Phi_{\mathbf{P}}(S) = \infty$, then $\text{h}_r(\mathbf{Q}_S)$ is either a linear set or a semilinear set. We will look at two cases, one where S is non-empty and one where S is the empty set. First let $S = \{\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_\ell\} \subseteq \Psi(b) \setminus \{\mathscr{B}\}$ be non-empty, let $\bar{v} = \sum_{1 \leq i \leq \ell} \Phi_{\mathbf{P}}(b) \, \text{h}_r(\mathscr{D}_i)$ and let

$$M = \{\bar{v} + a_1 \, \text{h}_r(\mathscr{D}_1) + a_2 \, \text{h}_r(\mathscr{D}_2) + \cdots + a_\ell \, \text{h}_r(\mathscr{D}_\ell) + a_{\ell+1} \, \text{h}_r(\mathscr{B}) \mid a_1, \ldots, a_\ell, a_{\ell+1} \in \mathbb{N}\}.$$

Clearly M is a linear set and we claim that $M = \text{h}_r(\mathbf{Q}_S)$. Let

$$\bar{u} = \bar{v} + u_1 \, \text{h}_r(\mathscr{D}_1) + u_2 \, \text{h}_r(\mathscr{D}_2) + \cdots + u_\ell \, \text{h}_r(\mathscr{D}_\ell) + u_{\ell+1} \, \text{h}_r(\mathscr{B}) \in M$$

for some $u_1, \ldots, u_\ell, u_{\ell+1} \in \mathbb{N}$ and let $\mathscr{D}$ be the σ-db with exactly $\Phi_{\mathbf{P}}(b) + u_1$ connected components isomorphic to $\mathscr{D}_1$, $\Phi_{\mathbf{P}}(b) + u_2$ connected components isomorphic to $\mathscr{D}_2$, ..., $\Phi_{\mathbf{P}}(b) + u_\ell$ connected components isomorphic to $\mathscr{D}_\ell$, $u_{\ell+1}$ connected components isomorphic to $\mathscr{B}$ and no other connected components. Clearly $\bar{u} = \mathrm{h}_r(\mathscr{D})$. Then since $\mathbf{P}$ is monotone and $\Phi_{\mathbf{P}}(S) = \infty$, $\mathscr{D} \in \mathbf{P}$ and so by the definition of $\mathbf{Q}_S$, $\mathscr{D} \in \mathbf{Q}_S$. On the other hand let $\mathscr{D} \in \mathbf{Q}_S$ then by definition for some $u_1, \ldots, u_\ell, u_{\ell+1} \in \mathbb{N}$, $\mathscr{D}$ contains exactly $\Phi_{\mathbf{P}}(b) + u_1$ connected components isomorphic to $\mathscr{D}_1$, $\Phi_{\mathbf{P}}(b) + u_2$ connected components isomorphic to $\mathscr{D}_2$, ..., $\Phi_{\mathbf{P}}(b) + u_\ell$ connected components isomorphic to $\mathscr{D}_\ell$, $u_{\ell+1}$ connected components isomorphic to $\mathscr{B}$ and no other connected components. The r-histogram vector of $\mathscr{D}$ is then

$$\bar{v} + u_1 \mathrm{h}_r(\mathscr{D}_1) + u_2 \mathrm{h}_r(\mathscr{D}_2) + \cdots + u_\ell \mathrm{h}_r(\mathscr{D}_\ell) + u_{\ell+1} \mathrm{h}_r(\mathscr{B})$$

and hence $\mathrm{h}_r(\mathscr{D}) \in M$. Therefore $M = \mathrm{h}_r(\mathbf{Q}_S)$. Now let $S = \emptyset$. Then $\mathbf{Q}_S$ contains only the σ-dbs in $\mathbf{Q}$ with connected components only isomorphic to $\mathscr{B}$. If $\Phi_{\mathbf{P}}(\{\mathscr{B}\}) = \infty$, then any σ-db with connected components only isomorphic to $\mathscr{B}$ is in $\mathbf{P}$ and hence $\mathrm{h}_r(\mathbf{Q}_S) = \{\bar{v} + a_1 \mathrm{h}_r(\mathscr{B}) \mid a_1 \in \mathbb{N}\}$ where $\bar{v}$ contains only 0's. Now if $\Phi_{\mathbf{P}}(\{\mathscr{B}\}) = g < \infty$, let $\mathscr{B}_i$ be the σ-db with exactly i connected components isomorphic to $\mathscr{B}$ and no other connected components. Then for every i where $0 \leq i < g$, $\mathscr{B}_i \in \mathbf{P}$ and for any $i \geq g$, $\mathscr{B}_i \notin \mathbf{P}$. Hence, $\mathrm{h}_r(\mathbf{Q}_S) = \bigcup_{0 \leq i < g} \{\mathrm{h}_r(\mathscr{B}_i)\}$ which is a semilinear set. We have proven that for every $S \subseteq \Psi(b) \setminus \{\mathscr{B}\}$, $\mathrm{h}_r(\mathbf{Q}_S)$ is either empty or a linear or semilinear set. Hence $\mathrm{h}_r(\mathbf{Q})$ is semilinear. □

Note that in the BDRD model we cannot obtain a similar result to Lemma 5.33 for hyperfinite hereditary properties. This is due to being only able to compare databases on the same number of elements in the BDRD model.

Combining Lemma 5.33 and Theorems 5.7 and 5.32 we obtain an alternative proof of the constant time testability (in the BDRD model) of any monotone hyperfinite property $\mathbf{P} \subseteq \mathbf{C}$, where $\mathbf{C}$ is closed under removing tuples.

Chapter 6

Towards approximate query enumeration with sublinear preprocessing time

In this chapter we introduce a new model for approximate query enumeration on classes of relational databases of bounded degree. We first prove that on databases of bounded degree any *local* first-order definable query can be enumerated approximately with constant delay after a preprocessing phase with *constant* running time. We extend this, showing that on databases of bounded tree-width and bounded degree, every query that is expressible in first-order logic can be enumerated approximately with constant delay after a preprocessing phase with *sublinear* (more precisely, *polylogarithmic*) running time.

Durand and Grandjean [29] proved that *exact* enumeration of first-order queries on databases of bounded degree can be done with constant delay after a preprocessing phase with running time *linear* in the size of the input database. Hence we achieve a significant speed-up in the preprocessing phase. Since sublinear running time does not allow reading the whole input database even once, sacrificing some accuracy is inevitable for our speed-up. Nevertheless, our enumeration algorithm comes with the following guarantees: With high probability, (1) only tuples are enumerated that are answers to the query or 'close' to being answers to the query, and (2) if the proportion of tuples that are answers to the query is sufficiently large, then all answers will be enumerated. For local first-order queries, only actual answers are enumerated, strengthening (1). Moreover, both the 'closeness' and the proportion required in (2) are controllable. Our algorithms only access the input database by sampling local parts, in a distributed fashion.

While our preprocessing phase is simpler than the preprocessing phase for the exact algorithm, our enumeration phase is more involved, as we push parts of the computation into the enumeration phase, allowing us to keep on enumerating answers.

We combine methods from property testing of bounded degree graphs with logic and query enumeration, which we believe can inspire further research.

We start this chapter, in Section 6.1, by introducing some notions and our running example which will be used throughout the chapter. In Section 6.2 we give some useful normal forms of first-order queries along with some results on local first-order queries. In Sections 6.3 and 6.4 we prove our main theorems on the enumeration of local and general first-order queries respectively. In Section 6.5, in an attempt to push the boundaries further, we prove strengthened versions of the theorems proved in Sections 6.3 and 6.4, showing how the required answer threshold can be reduced. In Section 6.6 we prove a generalisation of our main theorem on approximate enumeration of general first-order queries showing that the assumption of bounded tree-width can be replaced with the weaker assumption of Hanf-sentence testability. We also provide results on approximate membership testing and approximate counting. Finally, in Section 6.7, we prove that if we use the distance measure of the $\mathrm{BDRD}_{+/-}$ model (introduced in Chapter 5), we can obtain approximate enumeration algorithms for general first-order queries that have constant (rather than polylogarithmic) preprocessing time and constant delay.

6.1 Preliminaries

6.1.1 Enumeration problems

Let σ be a schema, let $\mathbf{C}$ be a class of σ-dbs and let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$. The *enumeration problem of ϕ over* $\mathbf{C}$ denoted by $\mathrm{Enum}_{\mathbf{C}}(\phi)$ is, given a database $\mathscr{D} \in \mathbf{C}$, to output the elements of $\phi(\mathscr{D})$ one by one with no repetition. An *enumeration algorithm* for the enumeration problem $\mathrm{Enum}_{\mathbf{C}}(\phi)$ with input database $\mathscr{D} \in \mathbf{C}$ proceeds in two phases, a preprocessing phase and an enumeration phase. The enumeration phase outputs all the elements of $\phi(\mathscr{D})$ with no duplicates. Furthermore, the enumeration phase has full access to the output of the preprocessing phase but can use only a constant total amount of extra memory.

The *delay* of an enumeration algorithm is the maximum time between the start of the enumeration phase and the first output (or the 'end of enumeration message' if there are no answers), two consecutive outputs, and the last output and the 'end of enumeration message'.

We focus on data complexity, i.e. we regard the query as being fixed, and the database being the input.

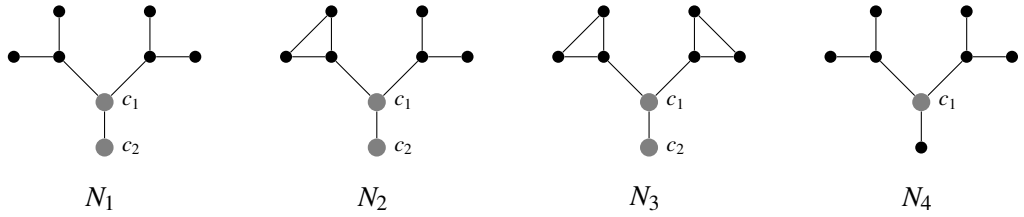

Fig. 6.1 The four 2-types of Examples 6.1 and 6.16. The vertices labelled 'c_1' and 'c_2' are the centres.

6.1.2 Local and non-local first-order queries

Let σ be a schema. We call an FO$[\sigma]$ formula $\phi(\bar{x})$ (with k free variables) *local* if there exists some $r \in \mathbb{N}$ such that for any σ-dbs $\mathscr{D}_1$ and $\mathscr{D}_2$ and tuples $\bar{a}_1 \in D_1^k$ and $\bar{a}_2 \in D_2^k$, if $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$ then, $\mathscr{D}_1 \models \phi(\bar{a}_1)$ if and only if $\mathscr{D}_2 \models \phi(\bar{a}_2)$. We call r the *locality radius of* ϕ. If an FO formula is not local we say it is *non-local*. We highlight that this notion of locality differs from that of Hanf locality and Gaifman locality of FO and should not be confused.

Proviso. *For the rest of the chapter, we fix a schema σ and numbers $d,t \in \mathbb{N}$ with $d \geq 2$. All databases are σ-dbs and have degree at most d, unless stated otherwise. We use $\mathbf{G}_d$ to denote the class of all graphs with degree at most d, $\mathbf{C}_d$ to denote the class of all σ-dbs with degree at most d, $\mathbf{C}_d^t$ to denote the class of all σ-dbs with degree at most d and tree-width at most t, and finally, we use $\mathbf{C}$ to denote a class of σ-dbs with degree at most d (which is closed under isomorphism).*

6.1.3 Running example

The following example is the basis of our running example which will be used throughout this chapter.

Example 6.1. On the class $\mathbf{G}_d$, consider the isomorphism types τ_2 and τ_4 of the 2-neighbourhoods $(N_2,(c_1,c_2))$ and $(N_4,(c_1))$ where N_2 and N_4 are the graphs shown in Figure 6.1 with centres (c_1,c_2) and (c_1). Let ϕ be the FO$[\{E\}]$-formula $\phi = \exists x \exists y \, \mathrm{sph}_{\tau_2}(x,y) \wedge \neg \exists z \, \mathrm{sph}_{\tau_4}(z)$. Let $\mathbf{P}$ be the property defined by ϕ on $\mathbf{G}_d$. We show that on the class $\mathbf{G}_d$, $\mathbf{P}$ is uniformly testable with constant time. For this, let $\varepsilon \in (0,1]$. Given oracle access to a graph $\mathscr{G} \in \mathbf{G}_d$ and $|V(\mathscr{G})| = n$ as an input, the ε-tester for $\mathbf{P}$ on $\mathbf{G}_d$ proceeds as follows:

1. If $n < 24d^3/\varepsilon$, do a full check of $\mathscr{G}$ and decide if $\mathscr{G} \in \mathbf{P}$.

2. Otherwise, uniformly and independently sample $\alpha = \log_{1-\varepsilon d/3} 1/3$ vertices from $[n]$.

3. For each sampled vertex, compute its 2-neighbourhood.

4. If a vertex is found with 2-type τ_4 then the tester rejects. Otherwise it accepts.

Claim 6.2. *The above ε-tester accepts with probability at least $2/3$ if $\mathscr{G} \in \mathbf{P}$ and rejects with probability at least $2/3$ if $\mathscr{G}$ is ε-far from $\mathbf{P}$. Furthermore, the ε-tester has constant query complexity and runs in constant time.*

Proof: Note that in τ_4, every vertex has 1 or 3 neighbours. For showing correctness, first assume $\mathscr{G} \in \mathbf{P}$. Then the tester will always accept as there exists no vertex with 2-type τ_4.

Now assume $\mathscr{G}$ is ε-far from $\mathbf{P}$. Then at least εdn edge modifications are necessary to make $\mathscr{G}$ isomorphic to a graph in $\mathbf{P}$. If $n < 24d^3/\varepsilon$ then the tester will reject so assume otherwise. Inserting a copy of τ_2 requires at most $8(d+1)$ modifications (pick 8 vertices and remove all incident edges then add the 8 edges to make an isolated copy of τ_2). Removing an edge uv from $\mathscr{G}$ will change the 2-type of any vertex in the set $N_2^{\mathscr{G}}(u) \cap N_2^{\mathscr{G}}(v)$. Lemma 3.2 (a) of [18] states that $|N_2^{\mathscr{G}}(u)| \leq d^{2+1}$ and $|N_2^{\mathscr{G}}(v)| \leq d^{2+1}$. Therefore, $|N_2^{\mathscr{G}}(u) \cap N_2^{\mathscr{G}}(v)| \leq d^3$ and hence inserting a copy of τ_2 could add at most $8d^4$ many copies of τ_4. After inserting a copy of τ_2 we need to remove all copies of τ_4. Let $v \in V(\mathscr{G})$ be a vertex with 2-type τ_4. Let u be the neighbour of v with degree 1. If we remove the edge $uv \in E(\mathscr{G})$, v's 2-type is no longer τ_4. Note that v has exactly 2 neighbours and u has 0 neighbours in $\mathscr{G} \setminus uv$. Moreover, we claim that by removing uv, we have introduced no new vertices with 2-type τ_4. To see this, observe that deleting uv will only affect the 2-types of vertices in $N_1^{\mathscr{G}}(v)$. But each vertex $x \in N_1^{\mathscr{G}}(v)$ will have a vertex with exactly two neighbours in its 2-neighbourhood in $\mathscr{G} \setminus uv$. Hence the new 2-type of x is not τ_4. This shows that there are at least $\varepsilon dn - 8(d+1) - 8d^4$ vertices with 2-type τ_4 in $\mathscr{G}$. Since $n \geq 24d^3/\varepsilon$ and $d \geq 2$, we have $8(d+1) \leq 8d^4 \leq \varepsilon dn/3$. The probability that we sample a vertex with 2-type τ_4 is therefore at least $\varepsilon dn/3n = \varepsilon d/3$. Hence the probability that none of the α sampled vertices have 2-type τ_4 is at most $(1-\varepsilon d/3)^{\alpha} = 1/3$. Therefore with probability at least $2/3$ the tester rejects.

For the running time, if $n < 24d^3/\varepsilon$ then we can do a full check of the input graph and decide if it has the property in time only dependent on d, ε and ϕ. Otherwise, note that the tester samples only a constant number of vertices in (2), and for each of the sampled vertices, the tester needs to make a constant number of oracle queries only to calculate its 2-neighbourhood in (3) because the degree is bounded. It can then be checked in constant time whether a vertex has 2-type τ_4. Therefore the tester has constant query complexity and constant running time. ■

6.2 Properties of first-order queries on bounded degree

In this section, we will give some useful normal forms of FO queries. We will then give a characterisation and some results for local FO queries.

6.2.1 General first-order queries

We make use of the following lemma to simplify Boolean combinations of sphere-formulas. We will use this result to show we can write FO queries in a special type of Hanf normal form that groups the Hanf-sentences and the sphere-formulas in a convenient way. Recall that $T_r^{\sigma,d}(k)$ is the set of all r-types with k centres and degree at most d, over schema σ.

Lemma 6.3 ([18])**.** *Let $r,k,d \in \mathbb{N}$ with $k \geq 1$, $d \geq 2$ and let σ be a schema. For every Boolean combination $\phi(\bar{x})$ of sphere-formulas of degree at most d and radius at most r, there exists an $I \subseteq T_r^{\sigma,d}(k)$ such that $\phi(\bar{x})$ is d-equivalent to $\bigvee_{\tau \in I} \mathrm{sph}_\tau(\bar{x})$.*

Furthermore, given $\phi(\bar{x})$, the set I can be computed in time $\mathrm{poly}(\|\phi\|) \cdot 2^{(kd^{r+1})^{\mathcal{O}(\|\sigma\|)}}$.

In the following lemma, we show that we can write any FO query as a disjunction of conjunctions of a sphere-formula and a boolean combination of Hanf-sentences. This normal form will be used in Lemma 6.18.

Lemma 6.4. *Let $\phi(\bar{x}) \in \mathrm{FO}$ and $|\bar{x}| = k$. Let r be the Hanf locality radius of ϕ. For every $d \in \mathbb{N}$ with $d \geq 2$ there exists a computable, d-equivalent formula to ϕ of the form*

$$\chi(\bar{x}) = \bigvee_{i \in [m]} \left(\mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s \right) \tag{6.1}$$

for some $m \in \mathbb{N}$, where for all $i \in [m]$, τ_i is an r-type with k centres and ψ_i^s is a conjunction of Hanf-sentences and negated Hanf-sentences. For each $\phi(\bar{x}) \in \mathrm{FO}$, we fix such a d-equivalent formula to ϕ (so we can refer to the *d-equivalent formula of ϕ in the form (6.1)).*

Proof. From ϕ we can construct a formula in the required form as follows. Firstly, by Theorem 2.27 we construct a d-equivalent formula in Hanf normal form. Next, we write the resulting formula in disjunctive normal form to obtain a formula of the form

$$\chi(\bar{x})' = \bigvee_{i \in [l]} \left(\psi_i^f(\bar{x}) \wedge \psi_i^s \right)$$

for some $l \in \mathbb{N}$, where for $i \in [l]$, $\psi_i^f(\bar{x})$ is a conjunction of sphere-formulas and negated sphere-formulas and ψ_i^s is a conjunction of Hanf-sentences and negated Hanf-sentences.

Then, by Lemma 6.3, we can replace each $\psi_i^f(\bar{x})$ with a d-equivalent formula $\bigvee_{t\in\lambda_i} \mathrm{sph}_t(\bar{x})$ where λ_i is a set of r-types with k centres. Finally, we replace each $\bigvee_{t\in\lambda_i} \mathrm{sph}_t(\bar{x}) \wedge \psi_i^s$ with $\bigvee_{t_i\in\lambda_i}(\mathrm{sph}_{t_i}(\bar{x}) \wedge \psi_i^s)$. The resulting formula is in the required form. □

In Theorems 6.14 and 6.20, we reduce the minimum size of the answer set required to enumerate all answers to the query ϕ in our approximate enumeration algorithms. We show we only actually require an answer set of size γn^c, where $c := \mathrm{conn}(\phi, d)$ is the maximum number of connected components in the r-neighbourhood (where r is the Hanf-locality radius of ϕ) of an answer to ϕ. We define $\mathrm{conn}(\phi, d)$ below.

Definition 6.5 ($\mathrm{conn}(\phi, d)$)**.** *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ where $|\bar{x}| = k$ and let $\chi(\bar{x})$ be the formula in the form (6.1) of Lemma 6.4 that is d-equivalent to ϕ. We define* $\mathrm{conn}(\phi, d)$ *as the maximum number of connected components of the neighbourhood types that appear in the sphere-formulas of χ. Note that* $\mathrm{conn}(\phi, d) \leq k$.

Recall that we fix a formula χ in the form (6.1) of Lemma 6.4 for each FO formula ϕ, and hence $\mathrm{conn}(\phi, d)$ is well defined.

6.2.2 Local first-order queries

We will start by showing that for any local FO query ϕ we can compute a set of r-types T (where r is the locality radius) such that for any σ-db $\mathscr{D}$ and tuple $\bar{a}$, $\bar{a}$ is an answer to ϕ on $\mathscr{D}$ if and only if the r-type of $\bar{a}$ is in T.

Lemma 6.6. *There is an algorithm that, given a local query $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with k free variables and given the locality radius r of ϕ, computes a set of r-types T with k centres such that for any σ-db $\mathscr{D}$ and tuple $\bar{a} \in D^k$, $\bar{a} \in \phi(\mathscr{D})$ if and only if the r-type of $\bar{a}$ in $\mathscr{D}$ is in T.*

Proof. Let T be an empty list. For each r-type τ with k centres we do the following. Let $\mathscr{D}_\tau$ be a representative σ-db of τ where $\bar{c}$ is the centre tuple, then if $\mathscr{D}_\tau \models \phi(\bar{c})$ add τ to T. Then since ϕ is local and r is the locality radius of ϕ, for every σ-db $\mathscr{D}$ and tuple $\bar{a} \in D^k$, $\mathscr{D} \models \phi(\bar{a})$ if and only if the r-type of $\bar{a}$ in $\mathscr{D}$ is in T. □

Using the previous lemma we will show that for any local FO query, σ-db $\mathscr{D}$ and tuple $\bar{a}$ from $\mathscr{D}$ it can be decided in constant time whether $\bar{a}$ is an answer to ϕ on $\mathscr{D}$. We will use this when approximately enumerating answers to local FO queries.

Lemma 6.7. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ be a local query with k free variables. There is an algorithm that, given a σ-db $\mathscr{D}$ and a tuple $\bar{a} \in D^k$, decides whether $\bar{a} \in \phi(\mathscr{D})$ in constant time.*

Proof. Let r be the locality radius of ϕ. First let us compute the set of r-types T as in Lemma 6.6. We will then compute the r-type τ of $\bar{a}$ in $\mathscr{D}$. By Lemma 6.6 if $\tau \in T$ then $\bar{a} \in \phi(\mathscr{D})$ and if $\tau \notin T$ then $\bar{a} \notin \phi(\mathscr{D})$.

Since r does not depend on $\mathscr{D}$, the r-type of $\bar{a}$ in $\mathscr{D}$ can be computed in constant time. Furthermore, computing the set T does not depend on $\mathscr{D}$, and hence it can be decided in constant time whether $\bar{a} \in \phi(\mathscr{D})$. □

We will finish this section with the following characterisations of local FO queries. We do not make use of these characterisations but we include them to aid intuition.

Observation 6.8. Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$. Then ϕ is local if and only if ϕ is d-equivalent to a boolean combination of sphere-formulas.

Proof. Let $|\bar{x}| = k$. First let us assume that ϕ is d-equivalent to a FO formula χ that is a boolean combination of sphere-formulas. Let r be the Hanf locality radius of χ. Then since χ contains no Hanf-sentences, for any σ-dbs $\mathscr{D}_1$ and $\mathscr{D}_2$ and tuples $\bar{a}_1 \in D_1^k$ and $\bar{a}_2 \in D_2^k$, if $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$ then, $\mathscr{D}_1 \models \phi(\bar{a}_1)$ if and only if $\mathscr{D}_2 \models \phi(\bar{a}_2)$. Hence ϕ is local and r is the locality radius of ϕ.

Now let us assume that ϕ is local. Let T be the set of r-types as constructed in Lemma 6.6. Therefore ϕ is d-equivalent to the formula $\bigvee_{\tau \in T} \mathrm{sph}_\tau(\bar{x})$ which is in the required form. □

Observation 6.9. For any local FO query ϕ, the locality radius of ϕ is equal to the Hanf locality radius of ϕ. Therefore, since the Hanf locality radius of an FO query is computable by Theorem 2.27, the locality radius of a local FO query is also computable.

Proof. Let ϕ be a local FO formula and let $r \in \mathbb{N}$ be the Hanf locality radius of ϕ. Since ϕ is local there exists an $r' \in \mathbb{N}$ such that for any σ-dbs $\mathscr{D}'_1$ and $\mathscr{D}'_2$ and tuples $\bar{a}'_1 \in D'^k_1$ and $\bar{a}'_2 \in D'^k_2$, if $\mathscr{N}_{r'}^{\mathscr{D}'_1}(\bar{a}'_1) \cong \mathscr{N}_{r'}^{\mathscr{D}'_2}(\bar{a}'_2)$ then, $\mathscr{D}'_1 \models \phi(\bar{a}'_1)$ if and only if $\mathscr{D}'_2 \models \phi(\bar{a}'_2)$. We will prove that $r = r'$. Let $\chi(\bar{x}) = \bigvee_{i \in [m]} \left(\mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s \right)$ be the formula that is d-equivalent to ϕ and is in the form (6.1) of Lemma 6.4. Note that the Hanf-locality radius of χ is r. If χ contains no Hanf-sentences then clearly $r = r'$. So let us assume χ does contain Hanf-sentences (which is possible for example if one of the sentence parts ψ_i^s is unsatisfiable). Now let us group the conjunctions in χ that contain the same sphere-formula. Therefore we obtain a d-equivalent formula to χ of the form

$$\chi'(\bar{x}) = \bigvee_{i \in [m']} \left(\mathrm{sph}_{\tau_i}(\bar{x}) \wedge (\psi_{i_1}^s \vee \psi_{i_2}^s \vee \cdots \vee \psi_{i_{\ell_i}}^s) \right).$$

For a contradiction let us assume that $r \neq r'$. Therefore there exists some σ-dbs $\mathscr{D}_1$ and $\mathscr{D}_2$ and tuples $\bar{a}_1 \in D_1^k$ and $\bar{a}_2 \in D_2^k$ such that $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$ and exactly one

of $\mathscr{D}_1 \models \phi(\bar{a}_1)$ and $\mathscr{D}_2 \models \phi(\bar{a}_2)$ is true. Without loss of generality let $\mathscr{D}_1 \models \phi(\bar{a}_1)$ and $\mathscr{D}_2 \not\models \phi(\bar{a}_2)$. Since $\mathscr{D}_1 \models \phi(\bar{a}_1)$, $\mathscr{D}_1 \models \chi'(\bar{a}_1)$ and hence there exists some $i \in [m']$ such that $\mathscr{D}_1 \models \mathrm{sph}_{\tau_i}(\bar{a}_1) \wedge (\psi^s_{i_1} \vee \psi^s_{i_2} \vee \cdots \vee \psi^s_{i_{\ell_i}})$. If there are no Hanf-sentences in the i-th subformula of χ' then we immediately reach a contradiction as $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$ and τ_i has radius r (by the construction of χ'). So let us assume there are Hanf-sentences in the i-th subformula of χ'. Let us denote the sentence $(\psi^s_{i_1} \vee \cdots \vee \psi^s_{i_{\ell_i}})$ as ψ_i.

If $r' < r$, then since $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$ it must also be true that $\mathscr{N}_{r'}^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_{r'}^{\mathscr{D}_2}(\bar{a}_2)$. Hence, by the definition of r' we have that $\mathscr{D}_1 \models \phi(\bar{a}_1)$ if and only if $\mathscr{D}_2 \models \phi(\bar{a}_2)$, which is a contradiction. So let us assume that $r' > r$. Let us consider a σ-db $\mathscr{D}$ that contains k-tuples $\bar{b}_1$ and $\bar{b}_2$ such that $\mathscr{N}_{r'}^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_{r'}^{\mathscr{D}}(\bar{b}_1)$ and $\mathscr{N}_{r'}^{\mathscr{D}_2}(\bar{a}_2) \cong \mathscr{N}_{r'}^{\mathscr{D}}(\bar{b}_2)$ (which exists since we can just take the σ-db that is a disjoint union of the fixed representatives of the r'-types of $\bar{a}_1$ and $\bar{a}_2$). Note that since $r' > r$, we have $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}}(\bar{b}_1)$ and $\mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2) \cong \mathscr{N}_r^{\mathscr{D}}(\bar{b}_2)$. Furthermore, since $\mathscr{N}_r^{\mathscr{D}_1}(\bar{a}_1) \cong \mathscr{N}_r^{\mathscr{D}_2}(\bar{a}_2)$, we also have that $\mathscr{N}_r^{\mathscr{D}}(\bar{b}_1) \cong \mathscr{N}_r^{\mathscr{D}}(\bar{b}_2)$. By the definition of r' and since $\mathscr{D}_1 \models \phi(\bar{a}_1)$, $\mathscr{D} \models \phi(\bar{b}_1)$, and hence $\mathscr{D} \models \chi'(\bar{b}_1)$. Recall that $i \in [m']$ is such that $\mathscr{D}_1 \models \mathrm{sph}_{\tau_i}(\bar{a}_1) \wedge \psi_i$ and τ_i has radius r. Furthermore, χ' contains only one subformula which contains $\mathrm{sph}_{\tau_i}(\bar{x})$ and each sphere-formula in χ' has radius r by construction. Therefore, $\mathscr{D} \models \mathrm{sph}_{\tau_i}(\bar{b}_1) \wedge \psi_i$ and so $\mathscr{D} \models \psi_i$. By the definition of r' again and since $\mathscr{D}_2 \not\models \phi(\bar{a}_2)$, $\mathscr{D} \not\models \phi(\bar{b}_2)$ and hence $\mathscr{D} \not\models \chi'(\bar{b}_2)$. This implies that $\mathscr{D} \not\models \mathrm{sph}_{\tau_i}(\bar{b}_2) \wedge \psi_i$. However, $\mathscr{D} \models \mathrm{sph}_{\tau_i}(\bar{b}_2)$ since $\mathscr{N}_r^{\mathscr{D}}(\bar{b}_1) \cong \mathscr{N}_r^{\mathscr{D}}(\bar{b}_2)$, and so it must be the case that $\mathscr{D} \not\models \psi_i$ which is a contradiction. Hence $r = r'$. □

6.3 Enumerating answers to local first-order queries

Assume ϕ is a local FO query with k free variables and $\mathscr{D}$ is a σ-db, such that the set $\phi(\mathscr{D})$ is larger than a fixed proportion of all possible k-tuples, i. e. $|\phi(\mathscr{D})| \geq \mu|D|^k$ for some fixed $\mu \in (0,1)$. It is easy to construct an algorithm that enumerates the set $\phi(\mathscr{D})$ with amortized constant delay, i. e. the average delay between any two outputs is constant. For each tuple $\bar{a} \in D^k$ (processed in, say, lexicographical order), the algorithm tests if $\bar{a}$ is in $\phi(\mathscr{D})$ (which can be done in constant time by Lemma 6.7 as ϕ is local) and outputs $\bar{a}$ if $\bar{a} \in \phi(\mathscr{D})$. Since we are assuming that $|\phi(\mathscr{D})|$ is larger than a fixed proportion of all possible tuples, the overall running time of the algorithm is $\mathscr{O}(|D|^k)$ and hence the algorithm has constant amortized delay. In this section we prove that we can de-amortize this algorithm using random sampling.

We begin this section by showing that there exists a randomised algorithm that does the following. The input is a set V which is partitioned into two sets V_1 and V_2. We assume that the algorithm can test in constant time if a given element from V is in V_1 or V_2. After

a constant time preprocessing phase, the algorithm enumerates a set S of elements with $S \subseteq V_1$, with constant delay. Furthermore, we show that if $|V_1|$ is large enough then with high probability $S = V_1$. We then use this result to prove our main theorem of this section (Theorem 6.13) on the approximate enumeration of the answers to a local query. In Theorem 6.14 we show that the relative size of the answer set can be reduced whilst still guaranteeing that with high probability we enumerate all answers to the query.

Lemma 6.10. *Fix $\mu \in (0,1)$ and $\delta \in (0,1)$. There exists a randomised algorithm which does the following. The input is a set V which is partitioned into two sets V_1 and V_2. We assume that the algorithm is given access to the size of V and can decide in constant time whether a given element from V is in V_1 or V_2. The algorithm outputs a set $S \subseteq V_1$ such that if $|V_1| \geq \mu|V|$ then, with probability at least δ, $S = V_1$.*

The algorithm has constant preprocessing time and enumerates S with no duplicates and constant delay between any two consecutive outputs.

Proof. Let $|V| = n$ and let us assume that V comes with a linear order over its elements, or equivalently that $V = [n]$. If V does not come with a linear order over its elements then we use the linear order induced by the encoding of V. Let $q = \min((1-\mu(1-\mu))^2, (1-\delta)^2/9)$. The preprocessing phase proceeds as follows:

1. Initialise an array $\mathbf{B}$ of length n. The array $\mathbf{B}$ contains one entry for each element in $[n]$ and it is used to record sampled elements. For an element $a \in [n]$, the entry $\mathbf{B}[a]$ is 1 if a has previously been sampled and it is 0 otherwise.

2. Initialise an empty queue $\mathbf{Q}$, to store tuples to be enumerated.

As discussed in Section 2.2 an array of any size can be initialised in constant time using lazy initialisation and hence the preprocessing phase runs in constant time.

Moving on to the enumeration phase, between each output the algorithm will sample a constant number of elements as well as going through a constant number of the elements in $[n]$ in order. The enumeration phase proceeds as follows:

1. Sample $\alpha = \lceil \log_{1-\mu(1-\mu)} q \rceil$ many elements uniformly and independently from $[n]$ and let t be a list of these elements.

2. Add the next $\lceil 1/\mu^2 \rceil$ elements from $[n]$ to t. If there are less than $\lceil 1/\mu^2 \rceil$ elements remaining just add all the remaining elements to t.

3. For each element a in t, if $\mathbf{B}[a] = 1$, skip this element. Otherwise, set $\mathbf{B}[a] = 1$ and if a is in V_1 add a to $\mathbf{Q}$.

4. If $\mathbf{Q} \neq \emptyset$, output the next element from $\mathbf{Q}$; stop otherwise.

5. Repeat Steps 1-4 until there is no element to output in Step 4.

In Steps 1 and 2 a list of elements is created which is of constant size. For each element in this list, in Step 3, the algorithm can check whether it is in V_1 in constant time and the arrays $\mathbf{Q}$ and $\mathbf{B}$ can be read and updated in constant time. Hence, each enumeration step can be done in constant time. This concludes the analysis of the running time. We now prove correctness.

Clearly, no duplicates will be enumerated due to the use of the array $\mathbf{B}$ and the only elements enumerated are those that are in V_1. Let S be the set of elements that are enumerated. We need to show that with probability at least $2/3$ if $|V_1| \geq \mu|V|$, then $S = V_1$. In each enumeration step we take the next $\lceil 1/\mu^2 \rceil$ elements from $[n]$. Assuming $|V_1| \geq \mu|V|$, after $\lceil n \cdot \mu^2 \rceil \leq \lceil \mu|V_1| \rceil$ enumeration steps the algorithm will have checked every element in $[n]$ and therefore $S = V_1$. Let us find a bound on the probability that we do at least $\lceil \mu|V_1| \rceil$ enumeration steps. We will start with the following claim which we will use when finding a bound on the probability that we do at least $\lceil \mu|V_1| \rceil$ enumeration steps.

Claim 6.11. *For all $q \in [0,1)$ and $m \in \mathbb{N}_{\geq 1}$, $\prod_{i=1}^{m}(1-q^{\frac{i+1}{2}}) \geq 1-3q^{\frac{1}{2}}$.*

Proof: First let us prove that

$$\prod_{i=1}^{m}(1-q^{\frac{i+1}{2}}) \geq 1-q^{\frac{1}{2}}-q-q^{\frac{3}{2}}+q^{\frac{m+2}{2}}$$

by induction on m.

For the base case, let $m = 1$, then

$$\prod_{i=1}^{1}(1-q^{\frac{i+1}{2}}) = 1-q \geq 1-q^{\frac{1}{2}}-q-q^{\frac{3}{2}}+q^{\frac{3}{2}}$$

as required.

Now for the inductive step. Let us assume the claim is true for m and we shall show the claim is true for $m+1$. We have

$$\prod_{i=1}^{m+1}(1-q^{\frac{i+1}{2}}) = \Big(\prod_{i=1}^{m}(1-q^{\frac{i+1}{2}})\Big) \cdot (1-q^{\frac{m+2}{2}}) \quad \geq (1-q^{\frac{1}{2}}-q-q^{\frac{3}{2}}+q^{\frac{m+2}{2}})(1-q^{\frac{m+2}{2}})$$

by the inductive hypothesis.

$$(1-q^{\frac{1}{2}}-q-q^{\frac{3}{2}}+q^{\frac{m+2}{2}})(1-q^{\frac{m+2}{2}}) = 1-q^{\frac{1}{2}}-q-q^{\frac{3}{2}}+q^{\frac{m+3}{2}}+q^{\frac{m+4}{2}}+q^{\frac{m+5}{2}}-q^{m+2}$$

$$\geq 1 - q^{\frac{1}{2}} - q - q^{\frac{3}{2}} + q^{\frac{m+3}{2}},$$

as $q^{(m+4)/2} + q^{(m+5)/2} - q^{m+2} \geq 0$.

Therefore,

$$\prod_{i=1}^{m}(1 - q^{\frac{i+1}{2}}) \geq 1 - q^{\frac{1}{2}} - q - q^{\frac{3}{2}} + q^{\frac{m+2}{2}} \geq 1 - 3q^{\frac{1}{2}}$$

as required ■

Claim 6.12. *Assume that $|V_1| \geq \mu n$. The probability that at least $\lceil \mu |V_1| \rceil$ distinct elements from V_1 are enumerated is at least $1 - 3q^{\frac{1}{2}}$.*

Proof: We shall start by showing that for $j \in \mathbb{N}$, where $1 \leq j \leq \lceil \mu |V_1| \rceil$, the probability that at least j distinct elements from V_1 are enumerated is at least $\prod_{i=1}^{j}(1 - q^{(i+1)/2})$.

We shall prove this by induction on j. For the base case, let $j = 1$. If an element from V_1 is sampled in the first enumeration step, then at least one element from V_1 will be enumerated. An element that is in V_1 is sampled with probability

$$\frac{|V_1|}{n} \geq \frac{\mu n}{n} = \mu \geq \mu(1 - \mu).$$

The probability that out of the α elements sampled in the first enumeration step there is none from V_1 is at most $(1 - \mu(1-\mu))^{\alpha} \leq q$ as $\alpha = \lceil \log_{1-\mu(1-\mu)} q \rceil \geq \log_{1-\mu(1-\mu)} q$. Therefore with probability at least $1 - q$ at least one element from $|V_1|$ is enumerated and hence we have proved the base case.

For the inductive step, assume that the claim is true for j, where $1 \leq j < \lceil \mu |V_1| \rceil$, we will show it is true for $j + 1$. Let us assume j distinct elements from V_1 have already been enumerated, and a total of at least $(j+1)\alpha$ elements have been sampled (of which at least j are from V_1). The probability an element from V_1 that was not already enumerated is sampled is $(|V_1| - j)/n$. Therefore, the probability that exactly j unique elements from V_1 have been sampled is at most

$$\left(1 - \frac{|V_1| - j}{n}\right)^{(j+1)\alpha - j} < (1 - \mu(1-\mu))^{(j+1)\alpha - j},$$

as $|V_1| - j > |V_1| - \mu |V_1| \geq \mu n(1 - \mu)$. Then

$$(1 - \mu(1-\mu))^{(j+1)\alpha - j} \leq \frac{q^{j+1}}{(1 - \mu(1-\mu))^{j}} \leq \frac{q^{j+1}}{(q^{\frac{1}{2}})^{j}} = q^{\frac{j+2}{2}},$$

since $\alpha = \lceil \log_{1-\mu(1-\mu)} q \rceil \geq \log_{1-\mu(1-\mu)} q$ and since $q \leq (1 - \mu(1-\mu))^2$ (by the choice of q). Therefore, the probability that there are at least $j + 1$ elements from V_1 in these sampled

tuples is at least $1-q^{(j+2)/2}$. Then by the inductive hypothesis, the probability that at least $j+1$ elements from V_1 are enumerated is at least

$$\Big(\prod_{i=1}^{j}(1-q^{\frac{j+1}{2}})\Big)\cdot(1-q^{\frac{j+2}{2}}) = \prod_{i=1}^{j+1}(1-q^{\frac{i+1}{2}})$$

as required.

Finally, by Claim 6.11, the probability that at least $\lceil \mu|V_1|\rceil$ many distinct elements from V_1 are enumerated is at least

$$\prod_{i=1}^{\lceil \mu|\phi(\mathscr{D})|\rceil}(1-q^{\frac{i+1}{2}}) \geq 1-3q^{\frac{1}{2}}.$$

■

By Claim 6.12 the probability that $S=V_1$ if $|V_1|\geq \mu n$ is at least $(1-3q^{\frac{1}{2}})\geq\delta$ by the choice of q. This completes the proof. □

We now use Lemma 6.10 to prove the following theorem.

Theorem 6.13. *Let $\phi(\bar{x})\in \mathrm{FO}[\sigma]$ be a local query with k free variables and let $\gamma\in(0,1)$. There exists an algorithm that is given a σ-db $\mathscr{D}$ as an input, that after a constant time preprocessing phase, enumerates a set S (with no duplicates) with constant delay between any two consecutive outputs, such that:*

1. *$S\subseteq\phi(\mathscr{D})$, and*
2. *if $|\phi(\mathscr{D})|\geq\gamma|D|^k$ (i.e. the number of answers to the query is larger than a fixed fraction of the total possible number of answers), then with probability at least $2/3$, $S=\phi(\mathscr{D})$.*

Proof. Given a tuple $\bar{a}\in|D|^k$ we can test in constant time whether $\bar{a}\in\phi(\mathscr{D})$ or $\bar{a}\notin\phi(\mathscr{D})$ by Lemma 6.7. We can partition the set D^k into two sets based on whether a tuple is in $\phi(\mathscr{D})$ or not. Therefore the algorithm from Lemma 6.10 (with $\delta=2/3$, $\mu=\gamma$, $V=|D|^k$, $V_1=\phi(\mathscr{D})$ and $V_2=|D|^k\setminus\phi(\mathscr{D})$) meets the requirements in the theorem statement. □

In our algorithms, to achieve constant preprocessing time and constant delay we require the number of answers to the query to be some fixed fraction of the total possible number of answers. Otherwise, with high probability the algorithm would not sample an answer in the enumeration phase and the algorithm would stop.

It seems natural to expect that for queries occurring in practice, the elements of an answer tuple are within a small distance of each other in the input database (i. e. the r-neighbourhood

of the answer has few connected components). In such scenarios, we can strengthen our main theorem by reducing the number of answers required to output all answers to the query with high probability.

Theorem 6.14. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ be a local query with locality radius r and let $\gamma \in (0,1)$. Let $c := \mathrm{conn}(\phi, d)$, i.e the maximum number of connected components in the r-neighbourhood of a tuple $\bar{a} \in \phi(\mathscr{D})$ for any σ-db $\mathscr{D}$. There exists an algorithm that, given a σ-db $\mathscr{D}$ as input, after a constant time preprocessing phase enumerates a set S (with no duplicates) with constant delay between any two consecutive outputs, such that the following hold.*

1. *$S \subseteq \phi(\mathscr{D})$, and*
2. *if $|\phi(\mathscr{D})| \geq \gamma |D|^c$, then with probability at least $2/3$, $S = \phi(\mathscr{D})$.*

We defer the proof of Theorem 6.14 to Section 6.5.

6.4 Enumerating answers to general first-order queries

We now shift our focus to enumerating answers to general FO queries, now they can be non-local in the sense that we can not check if a tuple is an answer to the query by only looking at its neighbourhood. We are aiming at sublinear preprocessing time hence we cannot read the whole input database and therefore will need to sacrifice some accuracy. We allow our algorithms to enumerate 'close' answers as well as actual answers. We start this section by defining our notion of approximation before proving our main result.

6.4.1 Our notion of approximation

We will start by defining our notion of closeness.

Definition 6.15 (ε-close answers to FO queries)**.** *Let $\mathscr{D} \in \mathbf{C}$ be a σ-db and let $\varepsilon \in (0,1]$. Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ be a query with k free variables and Hanf locality radius r. A tuple $\bar{a} \in D^k$ is* ε-close to being an answer of ϕ on $\mathscr{D}$ and $\mathbf{C}$ *if $\mathscr{D}$ can be modified (with tuple insertions and deletions) into a σ-db $\mathscr{D}' \in \mathbf{C}$ with at most $\varepsilon d|D|$ modifications (i.e $\mathrm{dist}(\mathscr{D}, \mathscr{D}') \leq \varepsilon d|D|$) such that $\bar{a} \in \phi(\mathscr{D}')$ and the r-type of $\bar{a}$ in $\mathscr{D}'$ is the same as the r-type of $\bar{a}$ in $\mathscr{D}$.*

We denote the set of all tuples that are ε-close to being an answer of ϕ on $\mathscr{D}$ and $\mathbf{C}$ as $\phi(\mathscr{D}, \mathbf{C}, \varepsilon)$. Note that $\phi(\mathscr{D}) \subseteq \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$.

We shall illustrate Definition 6.15 in the following example.

Example 6.16. On the class $\mathbf{G}_d$, consider the isomorphism types τ_1, τ_2 and τ_3 of the 2-neighbourhoods $(N_1,(c_1,c_2))$, $(N_2,(c_1,c_2))$ and $(N_3,(c_1,c_2))$ shown in Figure 6.1. Let $\phi \in \mathrm{FO}[\{E\}]$ be given by $\phi(x,y) := \mathrm{sph}_{\tau_1}(x,y) \vee (\mathrm{sph}_{\tau_2}(x,y) \wedge \neg(\exists z \exists w\, \mathrm{sph}_{\tau_1}(z,w)))$. This formula might be useful in scenarios where ideally we want to return pairs of vertices with a specific 2-type τ_1 but if there is no such pair then returning vertex pairs with a similar 2-type will suffice.

Let $\mathscr{G} \in \mathbf{G}_d$ be a graph on n vertices and $\varepsilon \in (0,1]$. First observe that for any pair $(u,v) \in V(\mathscr{G})^2$ with 2-type τ_1, $(u,v) \in \phi(\mathscr{G})$ and hence $(u,v) \in \phi(\mathscr{G},\mathbf{G}_d,\varepsilon)$.

Assume $(u,v) \in V(\mathscr{G})^2$ has 2-type τ_2. Then $(u,v) \in \phi(\mathscr{G})$ if and only if $\mathscr{G}$ contains no vertex pair of 2-type τ_1. The pair (u,v) is in $\phi(\mathscr{G},\mathbf{G}_d,\varepsilon)$ if and only if $\mathscr{G}$ can be modified (with edge modifications) into a graph $\mathscr{G}' \in \mathbf{G}_d$ with at most εdn modifications such that $(u,v) \in \phi(\mathscr{G}')$ and the 2-type of (u,v) in $\mathscr{G}'$ is still τ_2.

For example if $\mathscr{G}$ is at distance at most $\varepsilon dn - 4d - 6$ (assuming that n is large enough such that $\varepsilon dn - 4d - 6 > 0$) from a graph $\mathscr{G}'' \in \mathbf{G}_d$ such that $\mathscr{G}'' \models \exists x \exists y\, \mathrm{sph}_{\tau_2}(x,y) \wedge \neg(\exists z \exists w\, \mathrm{sph}_{\tau_1}(z,w))$ then $(u,v) \in \phi(\mathscr{G},\mathbf{G}_d,\varepsilon)$. To see this let us assume that such a graph $\mathscr{G}''$ exists. Note that since $\mathbf{G}_d$ is closed under isomorphism we can assume that $\mathscr{G}$ and $\mathscr{G}''$ are on the same vertices. Then if (u,v) has 2-type τ_2 in $\mathscr{G}''$, $(u,v) \in \phi(\mathscr{G},\mathbf{G}_d,\varepsilon)$ since $\varepsilon dn - 4d - 6 \leq \varepsilon dn$. So let us assume that (u,v) does not have 2-type τ_2 in $\mathscr{G}''$. Let $(u_1,v_1) \in V(\mathscr{G}'')^2$ have 2-type τ_2 (we know one exists). Then we remove every edge that has u, v, u_1 or v_1 as an endpoint (there are at most $2d+3$ such edges), and then for each edge we removed we insert the same edge back in but swapping any endpoint u to u_1 and v to v_1 and vice versa (this requires at most $2(2d+3)$ many edge modifications in total). By doing this we have essentially just swapped the labels of the vertices u and u_1 and v and v_1. Hence in the resulting graph $\mathscr{G}'$, (u,v) has 2-type τ_2 and $\mathscr{G}'$ still contains no pair of vertices with 2-type τ_1. Therefore $(u,v) \in \phi(\mathscr{G}')$, and the distance between $\mathscr{G}$ and $\mathscr{G}'$ is at most $\varepsilon dn - 4d - 6 + 2(2d+3) = \varepsilon dn$.

Finally, for any pair $(u,v) \in V(\mathscr{G})^2$ with 2-type τ_3, $(u,v) \notin \phi(\mathscr{G})$ and $(u,v) \notin \phi(\mathscr{G},\mathbf{G}_d,\varepsilon)$ as for every $\mathscr{G}' \in \mathbf{G}_d$ there does not exist a pair with 2-type τ_3 that is in $\phi(\mathscr{G}')$.

The set $\phi(\mathscr{D},\mathbf{C},\varepsilon)$ contains all tuples that are ε-close to being answers to ϕ. A tuple $\bar{a} \in D^k$ is in $\phi(\mathscr{D},\mathbf{C},\varepsilon)$ if only a relatively small (at most εdn) number of modifications to $\mathscr{D}$ are needed to make $\bar{a}$ an answer to ϕ without changing $\bar{a}$'s neighbourhood type. This can be seen as a notion of *structural* approximation. One might be tempted to define $\phi(\mathscr{D},\mathbf{C},\varepsilon)$ differently, namely as the set of tuples that can be turned into an answer to ϕ on $\mathscr{D}$ (without necessarily preserving the neighbourhood type) with at most εdn modifications to $\mathscr{D}$. However, if $\phi(\mathscr{D}) \neq \emptyset$, say, $\bar{a} \in \phi(\mathscr{D})$, then we can turn any tuple $\bar{b} \in D^k$ into an answer

for ϕ on $\mathscr{D}$ with only a constant number of modifications. This can be done by exchanging $\bar{b}$'s r-neighbourhood with $\bar{a}$'s, for some r depending on ϕ. This is not meaningful.

Let χ be as in (6.1) of Lemma 6.4 for ϕ. Note that only tuples with a neighbourhood type that appears in χ can be in the set $\phi(\mathscr{D}, \mathbf{C}, \varepsilon)$. Nevertheless, the difference $|\phi(\mathscr{D}, \mathbf{C}, \varepsilon)| - |\phi(\mathscr{D})|$ can be unbounded. The following example demonstrates this.

Example 6.17. Let ϕ, τ_1 and τ_2 be as in Example 6.16. For $m \in \mathbb{N}_{\geq 1}$, let $\mathscr{G}_{1,m}$ be the graph that contains m disjoint copies of τ_2 and 1 disjoint copy of τ_1. Note that $\mathscr{G}_{1,m}$ has $n = 8(m+1)$ vertices. The graph $\mathscr{G}_{1,m}$ can be modified with one edge modification to form a graph which satisfies $\neg(\exists z \exists w \operatorname{sph}_{\tau_1}(z,w))$ without modifying the 2-type of any pair $(u,v) \in V(\mathscr{G}_{1,m})^2$ with 2-type τ_2 in $\mathscr{G}_{1,m}$. Therefore if $1 \leq \varepsilon dn$ then every pair $(u,v) \in V(\mathscr{G}_{1,m})^2$ with 2-type τ_2 is in $\phi(\mathscr{G}_{1,m}, \mathbf{G}_d, \varepsilon)$. Hence, assuming $1 \leq \varepsilon dn$ we have $|\phi(\mathscr{G}_{1,m}, \mathbf{G}_d, \varepsilon)| - |\phi(\mathscr{G}_{1,m})| = m + 1 - 1 = \Theta(n)$.

While $\phi(\mathscr{D}, \mathbf{C}, \varepsilon)$ is a *structural* approximation of $\phi(\mathscr{D})$, Example 6.17 illustrates that it may not be a *numerical* approximation. However, in scenarios where the focus lies on structural closeness, this might not be an issue.

We say that the problem $\mathrm{Enum}_{\mathbf{C}}(\phi)$ can be *solved approximately* with $\mathscr{O}(H(n))$ preprocessing time and constant delay for answer threshold function $f(n)$, if for every parameter $\varepsilon \in (0,1]$, there exists an algorithm, which is given oracle access to an input database $\mathscr{D} \in \mathbf{C}$ and $|D| = n$ as an input, that proceeds in two steps.

1. A preprocessing phase that runs in time $\mathscr{O}(H(n))$, and
2. an enumeration phase that enumerates a set S of distinct tuples with constant delay between any two consecutive outputs.

Moreover, we require that with probability at least $2/3$, $S \subseteq \phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$ and, if $|\phi(\mathscr{D})| \geq f(n)$, then $\phi(\mathscr{D}) \subseteq S$. The algorithm can make oracle queries of the form (R, i, j) as discussed in Section 2.6 which allows us to explore bounded radius neighbourhoods in constant time. We call such an algorithm an *ε-approximate enumeration algorithm*.

6.4.2 Main results

Before proving our main result of this section on the approximate enumeration of general first-order queries, we start by proving the following lemma. In this lemma, we show that for a given database $\mathscr{D}$ and FO query ϕ we can compute a set of neighbourhood types in polylogarithmic time, that with high probability only contains the neighbourhood types of tuples that are answers or close to being answers to ϕ on $\mathscr{D}$. To compute this set we write

ϕ in the form (6.1) as in Lemma 6.4 and then run property testers on the sentence parts to determine with high probability whether tuples with the corresponding r-type (the r-type that appears in the sphere-formula) are answers to ϕ on the input database or are far from being an answer to ϕ on the input database.

Lemma 6.18. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with $|\bar{x}| = k$ and Hanf locality radius r and let $\varepsilon \in (0,1]$. There exists an algorithm $\mathbb{A}_\varepsilon$, which, given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d^t$ as input along with $|D| = n$, computes a set T of r-types with k centres such that with probability at least $5/6$, for any $\bar{a} \in D^k$,*

1. *if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and*
2. *if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T.*

Furthermore, $\mathbb{A}_\varepsilon$ runs in polylogarithmic time.

Proof. If $n < 8k/\varepsilon$ then we do a full check of $\mathscr{D}$ and form the set T exactly. Otherwise, $\mathbb{A}_\varepsilon$ starts by computing the formula $\chi(\bar{x})$ that is d-equivalent to ϕ and is in the form (6.1) as in Lemma 6.4. Let m be as in Lemma 6.4. By Theorem 2.35, any sentence definable in FO is uniformly testable on $\mathbf{C}_d^t$ in polylogarithmic time. Hence for every $i \in [m]$ there exists an $\varepsilon/2$-tester that runs in polylogarithmic time and with probability at least $2/3$ accepts if the input satisfies $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and rejects if the input is $\varepsilon/2$-far from satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$. We can amplify this probability to $(5/6)^{1/m}$ by repeating the tester a constant number of times and we denote the resulting $\varepsilon/2$-tester as π_i. Next, $\mathbb{A}_\varepsilon$ computes the set T as follows.

1. Let $T = \emptyset$.
2. For each $i \in [m]$, run π_i with $\mathscr{D}$ as input, and if π_i accepts, then add τ_i to T.

By Lemma 6.4, $\chi(\bar{x})$ can be computed in constant time (only dependent on d, $\|\phi\|$ and $\|\sigma\|$). Moreover, each $\varepsilon/2$-tester π_i runs in polylogarithmic time. Since m is a constant, $\mathbb{A}_\varepsilon$ runs in polylogarithmic time.

It now only remains to prove correctness. Let $\bar{a} \in D^k$ and let τ be the r-type of $\bar{a}$ in $\mathscr{D}$. Let us assume that each π_i correctly accepts if $\mathscr{D}$ satisfies $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and correctly rejects if $\mathscr{D}$ is $\varepsilon/2$-far from satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$, which happens with probability at least $(5/6)^{(1/m)\cdot m} = 5/6$.

First let us assume that $\bar{a} \in \phi(\mathscr{D})$. We shall show that $\tau \in T$. Since $\mathscr{D} \models \phi(\bar{a})$, there exists at least one $i \in [m]$ such that $\mathscr{D} \models \mathrm{sph}_{\tau_i}(\bar{a}) \wedge \psi_i^s$ (as ϕ is d-equivalent to $\chi(\bar{x}) = \bigvee_{i \in [m]} \left(\mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s \right)$). Hence, $\mathscr{D} \models \exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and as we are assuming π_i correctly

accepted, the r-type τ_i will have been added to T. Since $\mathscr{D} \models \mathrm{sph}_{\tau_i}(\bar{a})$, $\tau_i = \tau$, and therefore $\tau \in T$.

Now let us assume that $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$. We shall show that $\tau \notin T$. For a contradiction let us assume that $\tau \in T$ and hence there must exist some $i \in [m]$ such that $\mathscr{D}$ is $\varepsilon/2$-close to satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ on $\mathbf{C}_d^t$ and $\tau_i = \tau$. By definition there exists a σ-db $\mathscr{D}' \in \mathbf{C}_d^t$ such that $\mathscr{D}' \models \exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and $\mathrm{dist}(\mathscr{D}, \mathscr{D}') \leq \varepsilon dn/2$. Since any property defined by a FO sentence on $\mathbf{C}_d^t$ is closed under isomorphism we can assume that $\mathscr{D}'$ can be obtained from $\mathscr{D}$ with at most $\varepsilon dn/2$ tuple modifications. If in $\mathscr{D}'$ the r-type of $\bar{a}$ is no longer τ then we can modify $\mathscr{D}'$ with at most $4dk$ tuple modifications into a σ-db $\mathscr{D}'' \in \mathbf{C}_d^t$ such that the r-type of $\bar{a}$ is τ in $\mathscr{D}''$ and $\mathscr{D}'' \cong \mathscr{D}'$ (and hence $\bar{a} \in \phi(\mathscr{D}'')$). To do this we choose a tuple $\bar{b}$ whose r-type is τ in $\mathscr{D}'$ and for any tuple that contains an element from $\bar{a}$ or $\bar{b}$, delete it and add back the same tuple but with the elements from $\bar{a}$ exchanged for the corresponding elements from $\bar{b}$ and vice versa. This requires at most $4dk$ tuple modifications. Hence $\mathrm{dist}(\mathscr{D}, \mathscr{D}'') \leq \varepsilon dn/2 + 4dk \leq \varepsilon dn$ if $n \geq 8k/\varepsilon$ (which we can assume as otherwise we do a full check of $\mathscr{D}$ and compute T exactly) and so by definition $\bar{a} \in \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$ which is a contradiction. Therefore $\tau \notin T$.

Hence with probability at least $5/6$, for every $\bar{a} \in D^k$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T. □

We now use Lemmas 6.10 and 6.18 to prove our main result of this section (Theorem 6.19).

Theorem 6.19. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ where $|\bar{x}| = k$. Then $\mathrm{Enum}_{\mathbf{C}_d^t}(\phi)$ can be solved approximately with polylogarithmic preprocessing time and constant delay for answer threshold function $f(n) = \gamma n^k$ for any parameter $\gamma \in (0,1)$.*

Proof. Let r be the Hanf locality radius of ϕ. Let $\mathscr{D} \in \mathbf{C}_d^t$ with $|D| = n$, let $\varepsilon \in (0,1]$ and let $\gamma \in (0,1)$. We shall construct an ε-approximate enumeration algorithm for $\mathrm{Enum}_{\mathbf{C}_d^t}(\phi)$ that has answer threshold function $f(n) = \gamma n^k$, polylogarithmic preprocessing time and constant delay.

In the preprocessing phase, the algorithm starts by running the algorithm from Lemma 6.18 on $\mathscr{D}$ to compute a set T of r-types with k centres. Then the algorithm from the proof of Lemma 6.10 with $\mu = \gamma$, $\delta = 5/6$, $V = D^k$, $V_1 = \{\bar{a} \in D^k \mid \text{the r-type of } \bar{a} \text{ in } \mathscr{D} \text{ is in } T\}$ and $V_2 = D^k \setminus V_1$ is run.

By Lemma 6.18, the set T is computed in polylogarithmic time. Hence as the preprocessing phase from the proof of Lemma 6.10 runs in constant time, the whole preprocessing phase runs in polylogarithmic time. By Lemma 6.10 there is constant delay between any

two consecutive outputs. This concludes the analysis of the running time. We now prove correctness.

Let S be the set of tuples enumerated. By Lemma 6.10 no duplicates are enumerated and $S \subseteq V_1 = \{\bar{a} \in D^k \mid \text{the r-type of } \bar{a} \text{ is in } T\}$. By Lemma 6.18, with probability at least $5/6$, for every $\bar{a} \in D^k$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T. Therefore with probability at least $5/6$, $\phi(\mathscr{D}) \subseteq V_1$ and $V_1 \subseteq \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$. Hence with probability at least $5/6 > 2/3$, $S \subseteq \phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$ as required. As previously discussed with probability at least $5/6$, $\phi(\mathscr{D}) \subseteq V_1$ (note that if $\phi(\mathscr{D}) \subseteq V_1$, then $|V_1| \geq |\phi(\mathscr{D})|$). If we assume that $\phi(\mathscr{D}) \subseteq V_1$ and $|\phi(\mathscr{D})| \geq \gamma n^k = \gamma |V|$, then $|V_1| \geq \gamma |V|$ and by Lemma 6.10 with probability at least $5/6$, $S = V_1$ and hence $\phi(\mathscr{D}) \subseteq S$. Therefore the probability that $\phi(\mathscr{D}) \subseteq S$ if $|\phi(\mathscr{D})| \geq \gamma n^k$ is at least $(5/6)^2 > 2/3$ as required. This completes the proof. □

As discussed in Section 6.3, it is natural for us to expect that for queries that occur in practice, the neighbourhood of the answer tuple has few connected components. We saw that for local FO queries, in such scenarios we can reduce the number of answers required to output all answers to the query with high probability (Theorem 6.14). The following theorem shows how we can reduce the answer threshold function for general FO queries.

Theorem 6.20. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ and let $c := \mathrm{conn}(\phi, d)$. Then the problem $\mathrm{Enum}_{\mathbf{C}_d^t}(\phi)$ can be solved approximately with polylogarithmic preprocessing time and constant delay for answer threshold function $f(n) = \gamma n^c$ for any parameter $\gamma \in (0,1)$.*

We defer the proof of Theorem 6.20 to Section 6.5.

6.5 Reducing the answer threshold function

Before we prove Theorems 6.14 and 6.20 we start with some definitions (which are based on those introduced by Kazana and Segoufin in [50]) and some lemmas.

For each type $\tau \in T_r^{\sigma,d}(k)$ we fix a representative for the corresponding r-type and fix a linear order among its elements (where, for technical reasons, the centre elements always come first). This way, we can speak of the first, second, ..., element of an r-type. Let $\mathscr{D}$ be a σ-db and let $\bar{a}$ be a tuple in $\mathscr{D}$ with r-type τ. For technical reasons, if there are multiple isomorphism mappings from the r-neighbourhood of $\bar{a}$ to the fixed representative of τ, we use the isomorphism mapping which is of smallest lexicographical order (recall that we assume that $\mathscr{D}$ comes with a linear ordering on its elements). The cardinality of τ, denoted as $|\tau|$, is the number of elements in its representative.

Let $\mathscr{D}$ be a σ-db and $\bar{a}$ be a tuple of elements from $\mathscr{D}$. We say that $\bar{a}$ is *r-connected* if the r-neighbourhood of $\bar{a}$ in $\mathscr{D}$ is connected.

Let $s \in \mathbb{N}$, let $F = (\alpha_2, \ldots, \alpha_m)$ be a sequence of elements from $[d^{s+1}]$ (recall that the maximum size of an s-neighbourhood is d^{s+1}), and let $\bar{x} = (x_1, \ldots, x_m)$ be a tuple. We write $\bar{x} = F(x_1)$ for the fact that, for $j \in \{2, \ldots, m\}$, x_j is the α_j-th element of the s-neighbourhood of x_1. We call each such F an *s-binding of* $\bar{x}$. Given s-type τ, we say that an s-binding F of $\bar{x}$ is *r-good for* τ if $F(x_1)$ is r-connected for every x_1 with type τ.

For a given tuple $\bar{x} = (x_1, \ldots, x_k)$, an r-split of $\bar{x}$ is a set of triples $C = \{(C_1, F_1, \tau_1), \ldots, (C_\ell, F_\ell, \tau_\ell)\}$ where for each $i \in [\ell]$

- $\emptyset \neq C_i \subseteq \bar{x}$, $C_i \cap C_j = \emptyset$ for $i \neq j \in [\ell]$ and $\bigcup_{1 \leq i \leq \ell} C_i = \{x_1, \ldots, x_k\}$,
- τ_i is a $3rk$-type with 1 centre, and
- $F_i = (\alpha_2, \ldots, \alpha_{|C_i|})$ is a $3rk$-binding of a tuple with $|C_i|$ elements such that for each $j \in \{2, \ldots, |C_i|\}$, $\alpha_j \in [|\tau_i|]$ and F_i is r-good for τ_i.

We write $\bar{x}^i$ to represent the variables from C_i, x^i_1 to represent the most significant variable from C_i (i.e the variable in C_i which appears first in the tuple $\bar{x}$), x^i_2 to represent the second most significant variable from C_i (i.e the variable in C_i which appears second in the tuple $\bar{x}$) and so on. We define the formula

$$\mathrm{Split}^C_r(\bar{x}) := \bigwedge_{1 \leq i \neq j \leq \ell} (N_r(\bar{x}^i) \cap N_r(\bar{x}^j) = \emptyset) \wedge \bigwedge_{(C_i, F_i, \tau_i) \in C} (\bar{x}^i = F_i(x^i_1) \wedge \mathrm{sph}_{\tau_i}(x^i_1)).$$

We let $S^{\sigma,d}_r(k)$ denote the set of r-splits of tuples with k elements for σ-dbs with degree at most d. We denote the cardinality of $S^{\sigma,d}_r(k)$ as $s(r,k)$.

Remark 6.21. For any $r, k \in \mathbb{N}$, σ-db $\mathscr{D}$ and tuple $\bar{a} \in D^k$ there exists exactly one r-split C such that $\mathscr{D} \models \mathrm{Split}^C_r(\bar{a})$.

Let $\mathscr{D}$ be a σ-db, let $r, k, c \in \mathbb{N}$ where $c \leq k$ and let C be an r-split for a tuple with k elements. For tuples $\bar{a} \in D^c$ and $\bar{b} \in D^k$ we say that $\bar{b}$ is *found* from $\bar{a}$ and C, if $c = |C|$, $\mathscr{D} \models \mathrm{Split}^C_r(\bar{b})$ and for every $i \in [c]$, the element b^i_1 (from $\bar{b}$) according to C, is equal to a_i. Intuitively, $\bar{a}$ consists of the most significant elements from $\bar{b}$ according to C.

Remark 6.22. Let $\mathscr{D}$ be a σ-db and let $r, k \in \mathbb{N}$. For any $\bar{b} \in D^k$ there exists exactly one r-split C (of a tuple with k elements) and tuple $\bar{a}$ from $\mathscr{D}$ such that $\bar{b}$ is found from $\bar{a}$ and C.

Lemma 6.23. *Let $r, k, c \in \mathbb{N}$ where $c \leq k$. There exists an algorithm which, given a σ-db $\mathscr{D}$, a tuple $\bar{a} \in D^c$ and an r-split C of a tuple with k elements as input, returns a tuple $\bar{b} \in D^k$*

that is found from $\bar{a}$ and C if one exists and returns false otherwise. Furthermore if such a $\bar{b}$ exists then it is unique.

The running time of the algorithm depends only on r, $|C|$, k, σ and d.

Proof. Let $\mathscr{D}$ be a σ-db, let $\bar{a} \in D^c$ and let C be an r-split of a tuple with k elements. The following algorithm returns a tuple $\bar{b} \in D^k$ that is found from $\bar{a}$ and C if one exists and returns false otherwise.

1. If $|C| \neq c$ or $\mathscr{D} \not\models \bigwedge_{(C_i,F_i,\tau_i)\in C} \mathrm{sph}_{\tau_i}(a_i)$ then return false.
2. For each $i \in [c]$, let $\bar{b}^i$ be the tuple whose first element is a_i such that $\mathscr{D} \models (\bar{b}^i = F_i(a_i))$. Then let $\bar{b}$ be the tuple found by combining all the $\bar{b}^i$ according to C.
3. If $\mathscr{D} \models \bigwedge_{1\leq i\neq j\leq c}(N_r(\bar{b}^i) \cap N_r(\bar{b}^j) = \emptyset)$, return $\bar{b}$. Otherwise, return false.

The $3rk$-neighbourhood of an element can be computed in time only dependent on r, k, σ and d. Hence Steps 1 and 2 run in time only dependent on r, k, σ, d and $|C|$ since each $\bar{b}^i$ can be found by exploring the $3rk$-neighbourhood of a_i. In Step 3, for every $i \in [c]$, $N_r(\bar{b}^i)$ can be computed in time only dependent on r, $|\bar{b}^i| \leq k$, σ and d and hence the running time of Step 3 depends only on r, k, σ, d and $|C|$ also. Therefore the overall running time of the algorithm depends only on r, k, σ, d and $|C|$ as required.

Assume a tuple $\bar{b}$ is returned by the above algorithm from C and $\bar{a}$. Then clearly $c = |C|$, $\mathscr{D} \models \mathrm{Split}_r^C(\bar{b})$ and each b_1^i according to C is equal to a_i. Therefore $\bar{b}$ is found from $\bar{a}$ and C.

Now assume that there does exist a tuple $\bar{b} \in D^k$ that is found from $\bar{a}$ and C. Then $\bar{b}$ is unique as there is only one way to choose each tuple $\bar{b}^i$ such that $\mathscr{D} \models (\bar{b}^i = F_i(a_i))$. Furthermore, it is easy to see that $\bar{b}$ will be outputted by the above algorithm. □

Lemma 6.24. *Let $T \subseteq T_r^{\sigma,d}(k)$. We can compute a set of r-splits S for $\bar{x} = (x_1,\ldots,x_k)$ such that the following holds: For any σ-db $\mathscr{D}$ and tuple $\bar{a} \in D^k$, $\mathscr{D} \models \bigvee_{\tau\in T} \mathrm{sph}_\tau(\bar{a})$ if and only if $\mathscr{D} \models \bigvee_{C\in S} \mathrm{Split}_r^C(\bar{a})$.*

Proof. The algorithm proceeds as follows. Let S be an empty set. For each possible r-split $C = \{(C_1,F_1,\tau_1),\ldots,(C_\ell,F_\ell,\tau_\ell)\}$ of the tuple $\bar{x}$ do the following. Let $\mathscr{D}_0$ be the disjoint union of the fixed representatives of each τ_i. Let $\bar{b} \in D_0^k$ be a tuple such that $\mathscr{D}_0 \models \mathrm{Split}_r^C(\bar{b})$ (note that such a tuple exists by the definition of an r-split). Then if $\bar{b}$'s r-type in $\mathscr{D}_0$ is in T, add C to S.

Towards correctness let $\mathscr{D}$ be a σ-db and let $\bar{a} \in D^k$. Let $C = \{(C_1,F_1,\tau_1),\ldots,(C_\ell,F_\ell,\tau_\ell)\}$ be the r-split such that $\mathscr{D} \models \mathrm{Split}_r^C(\bar{a})$ (note that C is unique by Remark 6.21). Let $\mathscr{D}'$ be the disjoint union of the fixed representatives of each $3rk$-type that appears in C and let $\bar{b} \in D'^k$ be a tuple such that $\mathscr{D}' \models \mathrm{Split}_r^C(\bar{b})$. It remains to show that $\mathscr{N}_r^{\mathscr{D}}(\bar{a}) \cong \mathscr{N}_r^{\mathscr{D}'}(\bar{b})$.

This completes the proof because by the construction of S, it implies that $C \in S$ if and only if the r-type of $\bar{a}$ in $\mathscr{D}$ is in T (i.e. $\mathscr{D} \models \bigvee_{\tau \in T} \mathrm{sph}_\tau(\bar{a})$ if and only if $\mathscr{D} \models \bigvee_{C \in S} \mathrm{Split}_r^C(\bar{a})$). Recall that we use a_j^i and b_j^i to represent the elements from $\bar{a}$ and $\bar{b}$ respectively that are the elements from C_i that appear j-th in the tuples $\bar{a}$ and $\bar{b}$ respectively. As $\mathscr{D} \models \mathrm{Split}_r^C(\bar{a})$ and $\mathscr{D}' \models \mathrm{Split}_r^C(\bar{b})$, by the definition of the formula $\mathrm{Split}_r^C(\bar{x})$, it follows that $\mathscr{N}_{3rk}^{\mathscr{D}}(a_1^i) \cong \mathscr{N}_{3rk}^{\mathscr{D}'}(b_1^i)$ for every $i \in [\ell]$. For every $i \in [\ell]$ and $j \in [|C_i|]$, a_j^i is at distance at most $(2r+1)(|C_i|-1) \leq (2r+1)(k-1) \leq 3rk - r$ from a_1^i in $\mathscr{D}$, and b_j^i is at distance at most $(2r+1)(|C_i|-1) \leq (2r+1)(k-1) \leq 3rk - r$ from b_1^i in $\mathscr{D}'$ (since each F_i is r-good for τ_i). Therefore for every $i \in [\ell]$, the r-neighbourhoods of $\bar{a}_i$ and $\bar{b}_i$ are contained in the $3rk$-neighbourhoods of a_1^i and b_1^i respectively and hence $\mathscr{N}_r^{\mathscr{D}}(\bar{a}_i) \cong \mathscr{N}_r^{\mathscr{D}'}(\bar{b}_i)$. Then since $N_r^{\mathscr{D}}(\bar{a}_i) \cap N_r^{\mathscr{D}}(\bar{a}_j) = \emptyset$ and $N_r^{\mathscr{D}'}(\bar{b}_i) \cap N_r^{\mathscr{D}'}(\bar{b}_j) = \emptyset$ (since $\mathscr{D} \models \mathrm{Split}_r^C(\bar{a})$ and $\mathscr{D}' \models \mathrm{Split}_r^C(\bar{b})$), it follows that $\mathscr{N}_r^{\mathscr{D}}(\bar{a}) \cong \mathscr{N}_r^{\mathscr{D}'}(\bar{b})$. □

Let us first prove Theorem 6.20.

Proof of Theorem 6.20. Let $\mathscr{D} \in \mathbf{C}_d^t$ with $|D| = n$, let $\varepsilon \in (0,1]$ and let $\gamma \in (0,1)$. We will construct an ε-approximate enumeration algorithm for $\mathrm{Enum}_{\mathbf{C}_d^t}(\phi)$ that has answer threshold function $f(n) = \gamma n^c$, polylogarithmic preprocessing time and constant delay.

In the preprocessing phase, the algorithm starts by running the algorithm from Lemma 6.18 on $\mathscr{D}$ to compute a set T of r-types with k centres (where r is the Hanf locality radius of ϕ). The algorithm then computes the set of r-splits S from T as in Lemma 6.24. An empty queue $\mathbf{Q}$ is then initialised which will store tuples to be outputted in the enumeration phase.

Let $V = \bigcup_{1 \leq i \leq c} D^i$. Let V_1 be the set that contains all $\bar{a} \in V$ such that there exists a $C \in S$ and $\bar{b} \in D^k$ where $\bar{b}$ is found from $\bar{a}$ and C. Finally let $V_2 = V \setminus V_1$. Note that by Lemma 6.23, given a tuple $\bar{a} \in V$ it can be decided in constant time whether $\bar{a} \in V_1$.

The algorithm from Lemma 6.10 is then run with $\mu = \gamma/(c \cdot s(r,k))$, $\delta = 4/5$ and V, V_1 and V_2 as defined above. Once the enumeration phase of the algorithm from Lemma 6.10 starts we do the following.

1. Each time a tuple $\bar{a}$ is enumerated from the algorithm from Lemma 6.10, for each $C \in S$: run the algorithm from Lemma 6.23 with $\bar{a}$ and C and if a tuple is returned add it to $\mathbf{Q}$.

2. If $\mathbf{Q} \neq \emptyset$, output the next tuple from $\mathbf{Q}$; stop otherwise.

3. Repeat Steps 1-2 until there is no tuple to output in step 2.

From Lemma 6.18 the set T can be computed in polylogarithmic time. The set S can be constructed in constant time as $|T|$ is a constant and the number of possible r-splits

for a k-tuple is also a constant. Then as the preprocessing phase from the algorithm from Lemma 6.10 runs in constant time the overall running time of the preprocessing phase is polylogarithmic.

In the enumeration phase, by Lemma 6.10 there is constant delay between the outputs of the tuples $\bar{a}$ used in Step 1. For every such tuple, by the definition of the set V_1, there exists at least one r-split in S that leads to a tuple being added to $\mathbf{Q}$. Then as $|S|$ is a constant and the algorithm from Lemma 6.23 runs in constant time, the enumeration phase has constant delay as required. This concludes the analysis of the running time. Let us now prove correctness.

By Lemma 6.10 in Step 1 of the enumeration phase no duplicate tuples $\bar{a}$ will be considered. Since for every tuple $\bar{b} \in D^k$ there exists exactly one r-split C and tuple $\bar{a}$ from $\mathscr{D}$ such that $\bar{b}$ is found from C and $\bar{a}$ (Remark 6.22), no duplicates will be enumerated.

Now let us assume that the set of r-types T were computed correctly (i.e. for any $\bar{a} \in D^k$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T) which happens with probability at least $5/6$ by Lemma 6.18. Let $\bar{b} \in D^k$ have r-type τ in $\mathscr{D}$ and let $C \in S_r^{\sigma,d}(k)$ be such that $\mathscr{D} \models \mathrm{Split}_r^C(\bar{b})$.

If $\bar{b} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, $\tau \notin T$ and hence by Lemma 6.24, $C \notin S$ and so $\bar{b}$ will not be enumerated. Therefore with probability at least $5/6$ only tuples from $\phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$ will be enumerated.

If $\bar{b} \in \phi(\mathscr{D})$, then $\tau \in T$ and hence by Lemma 6.24, $C \in S$. Let $\bar{a}$ be the tuple such that $\bar{b}$ is found from $\bar{a}$ and C. Note that as the maximum number of connected components in the r-neighbourhood of $\bar{b}$ in $\mathscr{D}$ is c, $|\bar{a}| \leq c$ and hence $\bar{a} \in V$. Then by definition $\bar{a} \in V_1$. Hence if every tuple from V_1 is considered in Step 1 of the enumeration phase, every tuple in $\phi(\mathscr{D})$ will be enumerated. By Lemma 6.10 with probability at least δ if $|V_1| \geq \mu|V|$, every tuple from V_1 will be considered in Step 1. We know that $|V_1| \geq |\phi(\mathscr{D})|/s(r,k)$ since every $\bar{a} \in V_1$ leads us to at most $|S| \leq s(r,k)$ many tuples from $\phi(\mathscr{D})$ (since by Lemma 6.23 for any r-split $C \in S$ there is at most one tuple that is found from $\bar{a}$ and C). If $|\phi(\mathscr{D})| \geq \gamma n^c$ then $|V_1| \geq \gamma n^c / s(r,k) \geq \gamma |V|/(c \cdot s(r,k)) = \mu|V|$ since $|V| = \sum_{i=1}^{c} n^i \leq cn^c$ and by the choice of μ. Hence if $|\phi(\mathscr{D})| \geq \gamma n^c$ with probability at least $\delta \cdot 5/6 = 2/3$ every tuple from $\phi(\mathscr{D})$ will be enumerated. This completes the proof. □

We now prove Theorem 6.14 which is similar to the proof of Theorem 6.20.

Proof of Theorem 6.14. First let us note that if ϕ is local then by Lemma 6.6 we can compute a set T of r-types (where r is the locality radius of ϕ) in constant time such that for any σ-db $\mathscr{D}$ and tuple $\bar{a}$ from $\mathscr{D}$, the r-type of $\bar{a}$ in $\mathscr{D}$ is in T if and only if $\bar{a} \in \phi(\mathscr{D})$.

Then to construct an algorithm as in the theorem statement we can just use the algorithm from the proof of Theorem 6.20 but change it in two ways. Firstly we allow the input class to

be any class of bounded degree σ-dbs and secondly, we construct T as discussed above. The only part of the algorithm from the proof of Theorem 6.20 that runs in non-constant time is the construction of T and hence our algorithm has the required running times.

To prove correctness first note that in the proof of Theorem 6.20 the only reason the input class was $\mathbf{C}_d^t$ was to allow the set T to be computed efficiently and with high probability correctly. Now T is computed exactly and since the algorithm will only enumerate tuples that have r-type in T, only tuples that are answers to the query for the input database will be enumerated as required. The proof of (2) from the theorem statement is then very similar to the last paragraph in the proof of Theorem 6.20 (the only difference is that now for local queries this happens with higher probability as T is computed exactly every time). □

6.6 Further results

In this section, we start by generalising our result on approximate enumeration of general FO queries (Theorem 6.20). We identify a condition that we call *Hanf-sentence testability*, which is a weakening of the bounded tree-width condition, under which we still get approximate enumeration algorithms with the same probabilistic guarantees as before. Finally, we discuss approximation versions of query membership testing and counting.

6.6.1 Weakening the conditions for approximate enumeration

We first introduce Hanf-sentence testability, which is based on the Hanf normal form of a formula. It allows us to compute the set of r-types as in Lemma 6.18 efficiently. Theorem 6.28 below is the generalisation of Theorem 6.20, and Example 6.29 illustrates the use of this generalisation.

Definition 6.25 (Hanf-sentence testable)**.** *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ and $\chi(\bar{x})$ be the formula in the form (6.1) of Lemma 6.4 that is d-equivalent to ϕ. Let m be the number of conjunctive clauses in χ. We say that ϕ is* Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$ *if for every $i \in [m]$, the formula $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ is uniformly testable on $\mathbf{C}$ in time at most $H(n)$.*

We will illustrate Hanf sentence testability in the following example.

Example 6.26. Let ϕ be as in Example 6.16 and let $\mathscr{G} \in \mathbf{G}_d$. If there exists $(u,v) \in V(\mathscr{G})^2$ with 2-type τ_1 then there exists a vertex with 2-type τ_4 (where τ_4 is as in Example 6.1) and vice versa. Hence, ϕ can be easily transformed into the form (6.1) of Lemma 6.4 by replacing the subformula $\neg(\exists z \exists w\, \mathrm{sph}_{\tau_1}(z,w))$ with $\neg \exists^{\geq 1} z\, \mathrm{sph}_{\tau_4}(z)$. The resulting formula then has two conjunctive clauses, $sph_{\tau_2}(x,y) \wedge \neg \exists^{\geq 1} z\, \mathrm{sph}_{\tau_4}(z)$ and $sph_{\tau_1}(x,y)$. We saw in

Example 6.1 that $\exists x \exists y \, \mathrm{sph}_{\tau_2}(x,y) \wedge \neg \exists^{\geq 1} z \, \mathrm{sph}_{\tau_4}(z)$ is uniformly testable on $\mathbf{G}_d$ in constant time. The formula $\exists x \exists y \, \mathrm{sph}_{\tau_1}(x,y)$ is trivially testable in constant time on $\mathbf{G}_d$ since we can insert a copy of τ_1 into a graph $\mathscr{G} \in \mathbf{G}_d$ with at most $8d+7$ modifications and therefore if $8d+7 \leq \varepsilon d|V(\mathscr{G})|$ we can always accept and otherwise (i.e. if $|V(\mathscr{G})| < (8d+7)/\varepsilon d$) we can just do a full check of the graph for a copy of τ_1 in constant time. Hence, ϕ is Hanf-sentence testable on $\mathbf{G}_d$ in constant time.

Note that any FO query is Hanf sentence testable on $\mathbf{C}_d^t$ in polylogarithmic time. We will now prove a result that is similar to Lemma 6.18 but works for any class $\mathbf{C}$ and FO query ϕ where ϕ is Hanf-sentence testable on $\mathbf{C}$. This will then be used to show we can replace bounded tree-width with Hanf sentence testability and still obtain enumeration algorithms with the same probabilistic guarantees.

Lemma 6.27. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with $|\bar{x}| = k$ and Hanf locality radius r and let $\varepsilon \in (0,1]$. If ϕ is Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$ then there exists an algorithm $\mathbb{B}_\varepsilon$ that runs in time $\mathscr{O}(H(n))$, which, given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}$ as input along with $|D| = n$, computes a set T of r-types with k centres such that with probability at least $5/6$, for any $\bar{a} \in D^k$,*

1. *if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and*
2. *if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T.*

Proof. The algorithm $\mathbb{B}_\varepsilon$ is nearly identical to the algorithm $\mathbb{A}_\varepsilon$ from Lemma 6.18. The only difference being is we replace the input class $\mathbf{C}_d^t$ with $\mathbf{C}$. The $\varepsilon/2$-testers π_i used now have input class $\mathbf{C}$ (rather than $\mathbf{C}_d^t$) and as ϕ is Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$ each π_i runs in time $\mathscr{O}(H(n))$ (rather than polylogarithmic). Since all other parts of the algorithm $\mathbb{A}_\varepsilon$ run in constant time, it follows that $\mathbb{B}_\varepsilon$ runs in time $\mathscr{O}(H(n))$ as required. The proof of the correctness of $\mathbb{B}_\varepsilon$ is then identical to the proof of the correctness of $\mathbb{A}_\varepsilon$ (but with the input class $\mathbf{C}_d^t$ replaced with $\mathbf{C}$). □

We will now show that if a FO query ϕ is Hanf-sentence testable on a class $\mathbf{C}$ in time $H(n)$ then $\mathrm{Enum}_{\mathbf{C}}(\phi)$ can be solved approximately with preprocessing time $\mathscr{O}(H(n))$ and constant delay. Note we are still able to reduce the answer threshold function.

Theorem 6.28. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ and let $c := \mathrm{conn}(\phi, d)$. If ϕ is Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$, then $\mathrm{Enum}_{\mathbf{C}}(\phi)$ can be solved approximately with preprocessing time $\mathscr{O}(H(n))$ and constant delay for answer threshold function $f(n) = \gamma n^c$ for any $\gamma \in (0,1)$.*

Proof. Let $\varepsilon \in (0,1]$, let $\gamma \in (0,1)$ and let us assume that ϕ is Hanf-sentence testable on $\mathbf{C}$ in time $H(n)$. If we take the ε-approximate enumeration algorithm for $\mathrm{Enum}_{\mathbf{C}_d^t}(\phi)$ with answer threshold function $f(n) = \gamma n^c$ given in the proof of Theorem 6.20, which we will denote by $\mathbb{E}_{\phi,\mathbf{C}_d^t,\varepsilon}$, and make the following changes: replace the input class $\mathbf{C}_d^t$ with $\mathbf{C}$, and use Lemma 6.27 instead of Lemma 6.18 to compute the set of r-types T. Then we argue that the resulting algorithm $\mathbb{E}_{\phi,\mathbf{C},\varepsilon}$ is an ε-approximate enumeration algorithm for $\mathrm{Enum}_{\mathbf{C}}(\phi)$ with preprocessing time $\mathcal{O}(H(n))$ and constant delay for answer threshold function $f(n) = \gamma n^c$.

In the preprocessing phase of $\mathbb{E}_{\phi,\mathbf{C}_d^t,\varepsilon}$ the only part that runs in non-constant time is the construction of the set T (which takes polylogarithmic time). In $\mathbb{E}_{\phi,\mathbf{C},\varepsilon}$ it takes $\mathcal{O}(H(n))$ time to compute T and hence $\mathbb{E}_{\phi,\mathbf{C},\varepsilon}$ has preprocessing time $\mathcal{O}(H(n))$. Since $\mathbb{E}_{\phi,\mathbf{C}_d^t,\varepsilon}$ has constant delay, $\mathbb{E}_{\phi,\mathbf{C},\varepsilon}$ also has constant delay.

Since the only differences in Lemma 6.18 and Lemma 6.27 is the running times and the input class, the proof of the correctness of $\mathbb{E}_{\phi,\mathbf{C},\varepsilon}$ is the same as the proof of the correctness of $\mathbb{E}_{\phi,\mathbf{C}_d^t,\varepsilon}$ but with the input class $\mathbf{C}_d^t$ replaced with $\mathbf{C}$. □

We will now return to our running example where we discuss an FO query and input class, which previous theorems did not give us an approximate enumeration algorithm for, but by Theorem 6.28 can now be approximately enumerated.

Example 6.29. Let ϕ be the formula as in Example 6.16. We saw in Example 6.26 that ϕ is Hanf-sentence testable on $\mathbf{G}_d$ in constant time and that the formula in the form (6.1) of Lemma 6.4 that is d-equivalent to ϕ is $\chi(x,y) = \mathrm{sph}_{\tau_1}(x,y) \vee (\mathrm{sph}_{\tau_2}(x,y) \wedge \neg\exists^{\geq 1} z\, \mathrm{sph}_{\tau_4}(z))$. The maximum number of connected components of the neighbourhood types that appear in the sphere-formulas of χ is one. Hence, by Theorem 6.28, $\mathrm{Enum}_{\mathbf{G}_d}(\phi)$ can be solved approximately with constant preprocessing time and constant delay for answer threshold function $f(n) = \gamma n$ for any parameter $\gamma \in (0,1)$.

6.6.2 Approximate query membership testing

The *query membership testing problem* for $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ over $\mathbf{C}$ is the computational problem where, for a database $\mathscr{D} \in \mathbf{C}$, we ask whether a given tuple $\bar{a} \in D^k$ satisfies $\bar{a} \in \phi(\mathscr{D})$. We call $\bar{a}$ the *dynamical input* and the answer ('true' or 'false') the *dynamical answer*. Similar to query enumeration, the goal is to obtain an algorithm, that, after a preprocessing phase, can answer membership queries for dynamical inputs very efficiently. The preprocessing phase should also be very efficient. Kazana [49] shows that the query membership testing problem for any $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ over $\mathbf{C}$ can be solved by an algorithm with a linear time preprocessing phase, and an answering phase that, for a given dynamical input, computes the dynamical answer in constant time.

Given a local FO query, by Lemma 6.7, for any σ-db $\mathscr{D}$ and tuple $\bar{a}$ from $\mathscr{D}$ we can test in constant time whether $\bar{a} \in \phi(\mathscr{D})$. Hence in this section we shall focus on general FO queries.

We introduce an approximate version of the query membership testing problem. We say that the query membership testing problem for $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ over $\mathbf{C}$ can be *solved approximately* with an $\mathscr{O}(H(n))$-time preprocessing phase and constant time answering phase if for any $\varepsilon \in (0,1]$, there exists an algorithm, which is given oracle access to a database $\mathscr{D} \in \mathbf{C}$ and $|D| = n$ as an input, and proceeds in two phases.

1. A preprocessing phase that runs in time $\mathscr{O}(H(n))$.
2. An answer phase where, given dynamical input $\bar{a} \in D^k$, the following is computed in constant time.
   - If $\bar{a} \in \phi(\mathscr{D})$, the algorithm returns 'true', with probability at least $2/3$, and
   - if $\bar{a} \notin \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$, the algorithm returns 'false', with probability at least $2/3$.

The following follows from the proof of Lemma 6.18.

Theorem 6.30. *The query membership testing problem for $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ (where $|\bar{x}| = k$) over $\mathbf{C}_d^t$ can be solved approximately with a polylogarithmic preprocessing phase and constant time answering phase.*

Proof. Let r be the Hanf locality radius of ϕ. In the preprocessing phase a set T of r-types as in Lemma 6.18 is computed. Then in the answer phase, given a tuple $\bar{a} \in D^k$, the r-type τ of $\bar{a}$ is computed. If $\tau \in T$ then the algorithm returns 'true', otherwise it returns 'false'. By Lemma 6.18 the set T can be computed in polylogarithmic time and it takes constant time to calculate τ. By Lemma 6.18 with probability at least $5/6 > 2/3$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T. Therefore with probability at least $2/3$ if $\bar{a} \in \phi(\mathscr{D})$ the algorithm outputs 'true' and if $\bar{a} \notin \phi(\mathscr{D}, \mathbf{C}, \varepsilon)$ the algorithm outputs 'false' as required. □

Note that we can get a similar result for Hanf sentence testable FO queries over any class of bounded degree graphs.

6.6.3 Approximate counting

The *counting problem* for $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ over $\mathbf{C}$ is the problem of, given a database $\mathscr{D} \in \mathbf{C}$, compute $|\phi(\mathscr{D})|$. It was shown in [15] that the counting problem for any $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ over $\mathbf{C}$ can be solved in linear time.

Recall that EstimateFrequencies$_{r,k,s}$ is the algorithm, which is given oracle access to an input database $\mathscr{D} \in \mathbf{C}$, that samples s tuples from D^k uniformly and independently and explores their r-neighbourhoods. EstimateFrequencies$_{r,k,s}$ returns the distribution vector $\bar{v}$ of the r-types of this sample. EstimateFrequencies$_{r,k,s}$ has constant running time, independent of $|D|$. By Lemma 2.19 if

$$s \geq \frac{\mathrm{c}(r,k)^2}{\lambda^2} \cdot \ln\Big(\frac{2\,\mathrm{c}(r,k)}{1-\delta}\Big)$$

then with probability at least δ the vector $\bar{v}$ returned by EstimateFrequencies$_{r,k,s}$ on input $\mathscr{D}$ satisfies $\|\bar{v} - \mathrm{dv}_{r,k}(\mathscr{D})\|_1 \leq \lambda$.

By combining Lemmas 6.18 and 6.24 and Lemma 2.19 we get the following result.

Theorem 6.31. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with $|\bar{x}| = k$, let $\varepsilon \in (0,1]$, let $\lambda \in (0,1)$ and let $c := \mathrm{conn}(\phi, d)$. There exists an algorithm, which, given oracle access to $\mathscr{D} \in \mathbf{C}_d^t$ and $|D| = n$ as an input, returns an estimate of $|\phi(\mathscr{D})|$ such that with probability at least $2/3$ the estimate is within the range $[|\phi(\mathscr{D})| - \lambda cn^c, |\phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)| + \lambda cn^c]$. Furthermore, the algorithm runs in polylogarithmic time in n.*

Proof. Let r be the Hanf locality radius of ϕ, let $\delta = (9/10)^{1/c}$ and for every $i \in [c]$, let

$$s_i = \frac{\mathrm{c}(3rk,i)^2}{\lambda^2} \cdot \ln\Big(\frac{2\,\mathrm{c}(3rk,i)}{1-\delta}\Big).$$

The algorithm first runs EstimateFrequencies$_{3rk,i,s_i}$ on input $\mathscr{D}$ for every $i \in [c]$. Let $\bar{v}_1, \ldots, \bar{v}_c$ be the vectors returned.

Next, the algorithm computes a set T of r-types with k centres as in Lemma 6.18, and then computes a set of r-splits S from T as in Lemma 6.24. Let χ be the formula in the form (6.1) of Lemma 6.4. Note that since T contains only r-types that appear in χ, by Lemma 6.24 and the definition of the formula $\mathrm{Split}_r^C(\bar{x})$ every $C \in S$ satisfies $|C| \leq c$. Then for each $i \in [c]$, let $\bar{v}_i'$ be the vector with $c(3rk,i)$ components where the j-th component is the number of $C \in S$ such that in the fixed representative σ-db of the j-th $3rk$-type with i centres there exists a tuple that is found from C and the tuple of centres.

Then to estimate $|\phi(\mathscr{D})|$ the algorithm returns

$$\sum_{i \in [c]} \Big(\sum_{j \in [c(3rk,i)]} n^i \cdot v_i[j] \cdot v_i'[j] \Big).$$

Since c is a constant and each EstimateFrequencies$_{3rk,i,s_i}$ runs in constant time, it takes constant time to compute the vectors $\bar{v}_1, \ldots, \bar{v}_c$. By Lemma 6.18, T is computed in polylogarithmic time. Computing S does not depend on $\mathscr{D}$ and as each $c(3rk,i)$ and $|S|$ are

constants the tuples $\bar{v}'_i$ can be computed in constant time by Lemma 6.23. Finally, the estimate of $|\phi(\mathscr{D})|$ is computed from the vectors $\bar{v}_i$ and $\bar{v}'_i$ and hence the overall running time is polylogarithmic. Now let us prove correctness.

By Lemma 6.18 with probability at least $5/6$, for any $\bar{a} \in D^k$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi(\mathscr{D}, \mathbf{C}^t_d, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T. Therefore, with probability at least $5/6$ the number of tuples in $\mathscr{D}$ with r-type in T is within the range $[|\phi(\mathscr{D})|, |\phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}^t_d, \varepsilon)|$. Hence our aim is to estimate the number of tuples with r-type in T. We argue that the number of tuples in $\mathscr{D}$ with r-type in T is equal to

$$t = \sum_{i \in [c]} \Big(\sum_{j \in [c(3rk,i)]} n^i \cdot \mathrm{dv}_{3rk,i}(\mathscr{D})[j] \cdot v'_i[j] \Big).$$

Let $\bar{a} \in D^k$ have r-type τ in $\mathscr{D}$. Let C be the r-split such that $\mathscr{D} \models \bigvee_{C \in S} \mathrm{Split}^C_r(\bar{a})$ and let $\bar{b}$ be the tuple from $\mathscr{D}$ such that $\bar{a}$ is found from C and $\bar{b}$ (note that C and $\bar{b}$ are unique by Remark 6.22). Let τ' be the $3rk$-type of $\bar{b}$ in $\mathscr{D}$. First let us assume that $\tau \in T$. Then we will show that $\bar{a}$ will be counted exactly once in the sum t.

In the fixed representative σ-db of τ' there exists a tuple that is found from C and the tuple of centres (since the $3rk$-type of the i-th centre is equal to the $3rk$-type of b_i in $\mathscr{D}$). Therefore the value of $v'_{|\bar{b}|}[\tau']$ will be increased by one for C. Since there exists no other $C \in S$ and tuple $\bar{b}$ from $\mathscr{D}$ such that $\bar{a}$ is found from C and $\bar{b}$, $\bar{a}$ will be counted exactly once by t. Now assume $\tau \notin T$. We will show that $\bar{a}$ will not be counted by t. By the construction of S, $C \notin S$. Therefore the value of $v'_{|\bar{b}|}[\tau']$ will not be increased by one for C. Since there exists no other $C \in S$ and tuple $\bar{b}$ from $\mathscr{D}$ such that $\bar{a}$ is found from C and $\bar{b}$, $\bar{a}$ will not be counted by t. Therefore the number of tuples in $\mathscr{D}$ with r-type in T is equal to t.

By Lemma 2.19, with probability at least $(9/10)^{c/c} = 9/10$ for every $i \in [c]$, $\|\bar{v}_i - \mathrm{dv}_{3rk,i}(\mathscr{D})\|_1 \leq \lambda$. Hence with probability at least $9/10$, $\sum_{i \in [c]} \|\bar{v}_i - \mathrm{dv}_{3rk,i}(\mathscr{D})\|_1 \leq c\lambda$. Therefore by looking at the two extreme cases, with probability at least $9/10 \cdot 5/6 > 2/3$

$$|\phi(\mathscr{D})| - c\lambda n^c \leq \sum_{i \in [c]} \Big(\sum_{j \in [c(3rk,i)]} n^i \cdot v_i[j] \cdot v'_i[j] \Big) \leq |\phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}^t_d, \varepsilon)| + c\lambda n^c$$

as required. □

The obvious limitation of Theorem 6.31 is that $|\phi(\mathscr{D}) \cup \phi(\mathscr{D}, \mathbf{C}^t_d, \varepsilon)|$ can be much larger than $|\phi(\mathscr{D})|$, as discussed in Example 6.17. Nevertheless, in application where the focus is on structural closeness and very efficient running time, this might be tolerable.

6.7 Reducing the running time of the preprocessing phase to constant

In Chapter 5 we introduced the $\text{BDRD}_{+/-}$ model. We showed that in this model any property definable by a CMSO sentence (and hence by an FO sentence) on $\mathbf{C}_d^t$ is uniformly testable in constant time (Theorem 5.15). In this section, we will modify our definition of ε-close answers to FO queries to use the distance measure of the $\text{BDRD}_{+/-}$ model (i.e. allow inserting and deleting elements as well as tuples). We can then modify the proof of Lemma 6.18 to use Theorem 5.15 (rather than Theorem 2.35) to show that for a given database $\mathscr{D}$ and FO query ϕ we can compute a set of neighbourhood types in constant time, that with high probability only contains the neighbourhood types of tuples that are answers or close to being answers to ϕ on $\mathscr{D}$. This then allows us to solve $\text{Enum}_{\mathbf{C}_d^t}(\phi)$ approximately (in the $\text{BDRD}_{+/-}$ model) for any $\phi \in \text{FO}$ with constant (rather than polylogarithmic) preprocessing time and constant delay.

We will start by defining ε-close answers to FO queries in the $\text{BDRD}_{+/-}$ model.

Definition 6.32 (ε-close answers to FO queries in the $\text{BDRD}_{+/-}$ model)**.** *Let $\mathscr{D} \in \mathbf{C}$ be a σ-db and let $\varepsilon \in (0,1]$. Let $\phi(\bar{x}) \in \text{FO}[\sigma]$ be a query with k free variables and Hanf locality radius r. A tuple $\bar{a} \in D^k$ is* ε-close to being an answer of ϕ on $\mathscr{D}$ and $\mathbf{C}$ in the $\text{BDRD}_{+/-}$ model *if $\mathscr{D}$ can be modified (with tuple and element insertions and deletions) into a σ-db $\mathscr{D}' \in \mathbf{C}$ with at most $\varepsilon d \min\{|D|,|D'|\}$ modifications (i.e $\text{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \varepsilon d \min\{|D|,|D'|\}$) such that $\bar{a} \in \phi(\mathscr{D}')$ and the r-type of $\bar{a}$ in $\mathscr{D}'$ is the same as the r-type of $\bar{a}$ in $\mathscr{D}$.*

We denote the set of all tuples that are ε-close to being an answer of ϕ on $\mathscr{D}$ and $\mathbf{C}$ in the $\text{BDRD}_{+/-}$ model as $\phi_{+/-}(\mathscr{D},\mathbf{C},\varepsilon)$. Note that $\phi(\mathscr{D}) \subseteq \phi_{+/-}(\mathscr{D},\mathbf{C},\varepsilon)$.

We will illustrate Definition 6.32 in the following example.

Example 6.33. Let τ_1, τ_2, τ_3 and $\phi(x,y)$ be as in Example 6.16. Let $\mathscr{G} \in \mathbf{G}_d$ be a graph on n vertices and let $\varepsilon \in (0,1]$. First observe that for any pair $(u,v) \in V(\mathscr{G})^2$ with 2-type τ_1, $(u,v) \in \phi(\mathscr{G})$ and hence $(u,v) \in \phi_{+/-}(\mathscr{G},\mathbf{G}_d,\varepsilon)$.

Assume $(u,v) \in V(\mathscr{G})^2$ has 2-type τ_2. Then $(u,v) \in \phi(\mathscr{G})$ if and only if $\mathscr{G}$ contains no vertex pair of 2-type τ_1. The pair (u,v) is in $\phi_{+/-}(\mathscr{G},\mathbf{G}_d,\varepsilon)$ if and only if $\mathscr{G}$ can be modified (with edge and vertex modifications) into a graph $\mathscr{G}' \in \mathbf{G}_d$ with at most $\varepsilon d \min\{|V(\mathscr{G})|,|V(\mathscr{G}')|\}$ modifications such that $(u,v) \in \phi(\mathscr{G}')$ and the 2-type of (u,v) in $\mathscr{G}'$ is still τ_2.

For example if there exists a graph $\mathscr{G}'' \in \mathbf{G}_d$ such that $\mathscr{G}$ is at distance at most

$$\varepsilon d \min\{n,|V(\mathscr{G}'')|\} - 4d - 10$$

from $\mathscr{G}''$ (assuming that n is large enough such that $\varepsilon d \min\{n, |V(\mathscr{G}'')|\} - 4d - 10 > 0$) and such that $\mathscr{G}'' \models \exists x \exists y \operatorname{sph}_{\tau_2}(x,y) \wedge \neg(\exists z \exists w \operatorname{sph}_{\tau_1}(z,w))$ then $(u,v) \in \phi_{+/-}(\mathscr{G}, \mathbf{G}_d, \varepsilon)$. To see this let us assume that such a graph $\mathscr{G}''$ exists. Note that since $\mathbf{G}_d$ is closed under isomorphism we can assume that either $V(\mathscr{G}) \subseteq V(\mathscr{G}'')$ or $V(\mathscr{G}'') \subseteq V(\mathscr{G})$. Then if $(u,v) \in V(\mathscr{G}'')^2$ and (u,v) has 2-type τ_2 in $\mathscr{G}''$, $(u,v) \in \phi_{+/-}(\mathscr{G}, \mathbf{G}_d, \varepsilon)$ since $\varepsilon d \min\{n, |V(\mathscr{G}'')|\} - 4d - 10 \leq \varepsilon d \min\{n, |V(\mathscr{G}'')|\}$. So let us assume that either $(u,v) \notin V(\mathscr{G}'')^2$ or $(u,v) \in V(\mathscr{G}'')^2$ but (u,v) does not have 2-type τ_2 in $\mathscr{G}''$. Let $(u_1, v_1) \in V(\mathscr{G}'')^2$ have 2-type τ_2 (we know one exists). Then if $u \notin V(\mathscr{G}'')$, remove u_1 and insert u into $\mathscr{G}''$ and similarly if $v \notin V(\mathscr{G}'')$, remove v_1 and insert v into $\mathscr{G}''$. This requires at most 4 modifications. Then we remove every edge that has u, v, u_1 or v_1 as an endpoint (there are at most $2d+3$ such edges), and then for each edge we removed (including those which were removed as a result of removing either u_1 or v_1) we insert the same edge back in but swapping any endpoint u to u_1 and v to v_1 and vice versa (this requires at most $2(2d+3)$ many edge modifications in total). By doing this we have essentially just swapped the labels of the vertices u and u_1 and v and v_1. Hence in the resulting graph $\mathscr{G}'$, (u,v) has 2-type τ_2 and $\mathscr{G}'$ still contains no pair of vertices with 2-type τ_1. Therefore $(u,v) \in \phi(\mathscr{G}')$, and the distance between $\mathscr{G}$ and $\mathscr{G}'$ is at most $\varepsilon d \min\{n, |V(\mathscr{G}'')|\} - 4d - 10 + 2(2d+3) + 4 = \varepsilon d \min\{n, |V(\mathscr{G}')|\}$ since $|V(\mathscr{G}'')| = |V(\mathscr{G}')|$.

Finally, for any pair $(u,v) \in V(\mathscr{G})^2$ with 2-type τ_3, $(u,v) \notin \phi(\mathscr{G})$ and $(u,v) \notin \phi_{+/-}(\mathscr{G}, \mathbf{G}_d, \varepsilon)$ as for every $\mathscr{G}' \in \mathbf{G}_d$ there does not exist a pair with 2-type τ_3 that is in $\phi(\mathscr{G}')$.

We say that the problem $\operatorname{Enum}_{\mathbf{C}}(\phi)$ can be *solved approximately in the* $\mathrm{BDRD}_{+/-}$ *model* with $\mathscr{O}(H(n))$ preprocessing time and constant delay for answer threshold function $f(n)$, if for every parameter $\varepsilon \in (0,1]$, there exists an algorithm, which is given oracle access to an input database $\mathscr{D} \in \mathbf{C}$ and $|D| = n$ as an input, that proceeds in two steps.

1. A preprocessing phase that runs in time $\mathscr{O}(H(n))$, and
2. an enumeration phase that enumerates a set S of distinct tuples with constant delay between any two consecutive outputs.

Moreover, we require that with probability at least $2/3$, $S \subseteq \phi(\mathscr{D}) \cup \phi_{+/-}(\mathscr{D}, \mathbf{C}, \varepsilon)$ and, if $|\phi(\mathscr{D})| \geq f(n)$, then $\phi(\mathscr{D}) \subseteq S$.

Before proving the main result of this section on the approximate enumeration of FO queries in the $\mathrm{BDRD}_{+/-}$ model, we prove a similar lemma to Lemma 6.18 for the $\mathrm{BDRD}_{+/-}$ model. However, in the $\mathrm{BDRD}_{+/-}$ model we obtain an algorithm that runs in constant time rather than polylogarithmic.

Lemma 6.34. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ with $|\bar{x}| = k$ and Hanf locality radius r and let $\varepsilon \in (0,1]$. There exists an algorithm $\mathbb{A}_\varepsilon$, which, given oracle access to a σ-db $\mathscr{D} \in \mathbf{C}_d^t$ as input along with $|D| = n$, computes a set T of r-types with k centres such that with probability at least $5/6$, for any $\bar{a} \in D^k$,*

1. *if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and*
2. *if $\bar{a} \in D^k \setminus \phi_{+/-}(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T.*

Furthermore, $\mathbb{A}_\varepsilon$ runs in constant time.

Proof. If $n < 8k/\varepsilon$ then we do a full check of $\mathscr{D}$ and form the set T exactly. Otherwise, $\mathbb{A}_\varepsilon$ is similar to the algorithm given in Lemma 6.18. The algorithm $\mathbb{A}_\varepsilon$ starts by computing the formula $\chi(\bar{x})$ that is d-equivalent to ϕ and is in the form (6.1) as in Lemma 6.4. Let m be as in Lemma 6.4. By Theorem 5.15, any sentence definable in FO is uniformly testable on $\mathbf{C}_d^t$ in constant time (in the $\mathrm{BDRD}_{+/-}$ model). Hence for every $i \in [m]$ there exists an $\varepsilon/2$-tester that runs in constant time and with probability at least $2/3$ accepts if the input satisfies $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and rejects if the input is $\varepsilon/2$-far (in the $\mathrm{BDRD}_{+/-}$ model) from satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$. We can amplify this probability to $(5/6)^{1/m}$ by repeating the tester a constant number of times and we denote the resulting $\varepsilon/2$-tester as π_i. Next, $\mathbb{A}_\varepsilon$ computes the set T as follows.

1. Let $T = \emptyset$.
2. For each $i \in [m]$, run π_i with $\mathscr{D}$ as input, and if π_i accepts, then add τ_i to T.

By Lemma 6.4, $\chi(\bar{x})$ can be computed in constant time (only dependent on d, $\|\phi\|$ and $\|\sigma\|$). Moreover, each $\varepsilon/2$-tester π_i runs in constant time. Since m is a constant, $\mathbb{A}_\varepsilon$ runs in constant time.

It now only remains to prove correctness. Let $\bar{a} \in D^k$ and let τ be the r-type of $\bar{a}$ in $\mathscr{D}$. Let us assume that each π_i correctly accepts if $\mathscr{D}$ satisfies $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and correctly rejects if $\mathscr{D}$ is $\varepsilon/2$-far (in the $\mathrm{BDRD}_{+/-}$ model) from satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$, which happens with probability at least $(5/6)^{(1/m)\cdot m} = 5/6$.

First let us assume that $\bar{a} \in \phi(\mathscr{D})$. With the same arguments given in the proof of Lemma 6.18, $\tau \in T$.

Now let us assume that $\bar{a} \in D^k \setminus \phi_{+/-}(\mathscr{D}, \mathbf{C}_d^t, \varepsilon)$. We will show that $\tau \notin T$. For a contradiction let us assume that $\tau \in T$ and hence there must exist some $i \in [m]$ such that $\mathscr{D}$ is $\varepsilon/2$-close to satisfying $\exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ on $\mathbf{C}_d^t$ and $\tau_i = \tau$. By definition there exists a σ-db $\mathscr{D}' \in \mathbf{C}_d^t$ such that $\mathscr{D}' \models \exists \bar{x}\, \mathrm{sph}_{\tau_i}(\bar{x}) \wedge \psi_i^s$ and $\mathrm{dist}_{+/-}(\mathscr{D}, \mathscr{D}') \leq \varepsilon d \min\{|D|, |D'|\}/2$. Since any property defined by a FO sentence on $\mathbf{C}_d^t$ is closed under isomorphism we can

assume that $\mathscr{D}'$ can be obtained from $\mathscr{D}$ with at most $\varepsilon d \min\{|D|,|D'|\}/2$ tuple and element modifications. Note that $\bar{a}$ may not be in D'^k. If $\bar{a} \notin D'^k$ or in $\mathscr{D}'$ the r-type of $\bar{a}$ is no longer τ then we can modify $\mathscr{D}'$ with at most $4dk+2k$ tuple and element modifications into a σ-db $\mathscr{D}'' \in \mathbf{C}^t_d$ such that $\bar{a} \in D'^k$, the r-type of $\bar{a}$ is τ in $\mathscr{D}''$ and $\mathscr{D}'' \cong \mathscr{D}'$ (and hence $\bar{a} \in \phi(\mathscr{D}'')$). To do this we choose a tuple $\bar{b} \in D'^k$ whose r-type is τ in $\mathscr{D}'$. Then for each element a_i in $\bar{a}$, if $a_i \notin D'$, then insert a_i into $\mathscr{D}'$ and remove the corresponding element b_i of the tuple $\bar{b}$ from $\mathscr{D}'$. This requires at most $2k$ element modifications. We then remove every tuple that contains an element from $\bar{a}$ or $\bar{b}$, and for every tuple that was removed (including those which were removed as a result of removing an element of $\bar{b}$ from $\mathscr{D}$) add back the same tuple but with the elements from $\bar{a}$ exchanged for the corresponding elements from $\bar{b}$ and vice versa. This requires at most $4dk$ tuple modifications. Hence

$$\begin{aligned}\operatorname{dist}_{+/-}(\mathscr{D},\mathscr{D}'') &\leq \frac{\varepsilon d \min\{|D|,|D'|\}}{2} + 4dk + 2k \\ &\leq \frac{\varepsilon d \min\{|D|,|D'|\}}{2} + 6dk \\ &= \frac{\varepsilon d \min\{|D|,|D''|\}}{2} + 6dk\end{aligned}$$

since $|D'| = |D''|$. Furthermore, since $\operatorname{dist}_{+/-}(\mathscr{D},\mathscr{D}') \leq \varepsilon d \min\{|D|,|D'|\}/2$, we have $|D'| \geq |D| - \varepsilon d \min\{|D|,|D'|\}/2 \geq |D| - \varepsilon d|D'|/2$ and hence $|D''| = |D'| \geq |D|/(1+\varepsilon d/2)$. Then if $|D| \geq 12k(1+\varepsilon d/2)/\varepsilon$ (which we can assume as otherwise we do a full check of $\mathscr{D}$ and compute T exactly), $\min\{|D|,|D''|\} \geq 12k/\varepsilon$ and so $6dk \leq \varepsilon d \min\{|D|,|D''|\}/2$. Hence $\operatorname{dist}_{+/-}(\mathscr{D},\mathscr{D}'') \leq \varepsilon d \min\{|D|,|D''|\}$. Therefore by definition $\bar{a} \in \phi_{+/-}(\mathscr{D},\mathbf{C}^t_d,\varepsilon)$ which is a contradiction and so $\tau \notin T$.

Hence with probability at least $5/6$, for every $\bar{a} \in D^k$, if $\bar{a} \in \phi(\mathscr{D})$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is in T, and if $\bar{a} \in D^k \setminus \phi_{+/-}(\mathscr{D},\mathbf{C}^t_d,\varepsilon)$, then the r-type of $\bar{a}$ in $\mathscr{D}$ is not in T. □

Theorem 6.35. *Let $\phi(\bar{x}) \in \mathrm{FO}[\sigma]$ and let $c := \operatorname{conn}(\phi,d)$. Then the problem $\operatorname{Enum}_{\mathbf{C}^t_d}(\phi)$ can be solved approximately in the $\mathrm{BDRD}_{+/-}$ model with constant preprocessing time and constant delay for answer threshold function $f(n) = \gamma n^c$ for any parameter $\gamma \in (0,1)$.*

Proof. The proof of this Theorem is very similar to the proof of Theorem 6.20. The algorithm is the same except T is computed by Lemma 6.34 in constant time. That is the set T of neighbourhood types computed in the algorithm now, with high probability, contains all r-types of $\bar{a} \in \phi(\mathscr{D})$ but no r-types of $\bar{a} \in D^k \setminus \phi_{+/-}(\mathscr{D},\mathbf{C}^t_d,\varepsilon)$ (rather than $\bar{a} \in D^k \setminus \phi(\mathscr{D},\mathbf{C}^t_d,\varepsilon)$). Since the only step in the algorithm given in the proof of Theorem 6.20 that runs in non-constant time is the construction of T, our algorithm has constant

preprocessing time and constant delay as required. The proof of correctness is then identical to that given in the proof of Theorem 6.14. □

Chapter 7

Conclusion

7.1 Summary of contributions

In this thesis, we have studied existing models and introduced new models for the approximate evaluation of logically defined relational database queries. For boolean queries, we used the framework of property testing, where we used the bounded degree relational database model (the BDRD model) and introduced a new model, which is an extension of the BDRD model, called the $\text{BDRD}_{+/-}$ model. For non-boolean queries, we introduced a new model for the problem of approximate query enumeration on relational databases of bounded degree that can be seen as a relaxation of the classical exact version.

7.1.1 Approximate boolean query evaluation

In the BDRD model, in Chapter 3, we noted that the existential fragment of FO is (trivially) uniformly testable in constant time, and we proved that any property definable in the universal fragment of FO is also uniformly testable in constant time. Furthermore, in Chapter 5 we gave an alternative proof of the constant time uniform testability of hyperfinite monotone properties in the BDRD model.

In Chapter 5 we introduced our new model, the $\text{BDRD}_{+/-}$ model, where we also allow the insertion and deletion of elements in our distance measure. In this model, we showed that any property definable by a CMSO sentence is uniformly testable in constant time on databases of bounded degree and bounded tree-width (Theorem 5.15). This result improves the best known running time for such properties given in [4].

We actually proved two more general results than Theorem 5.15. First, we proved that any hyperfinite property whose histogram vectors form a semilinear set are uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model (Theorem 5.14). We then proved that being hyperfinite

and having a set of histogram vectors that are close to being semilinear is enough for constant time uniform testability in the $\text{BDRD}_{+/-}$ model (Theorem 5.19). We proved that hyperfinite hereditary properties are examples of such properties, i.e. they are close to being semilinear (Lemma 5.26). Combining Theorem 5.19 and Lemma 5.26 we get an alternative proof that hyperfinite hereditary properties are uniformly testable in constant time in the $\text{BDRD}_{+/-}$ model (Theorem 5.22). We believe that this highlights that semilinearity of neighbourhood histograms is a natural and powerful concept.

7.1.2 Constant size databases that preserve the local structure of large databases

On the way to proving our results in the $\text{BDRD}_{+/-}$ model, we partially answered an open question by Alon. Alon [55, Proposition 19.10] proved that on bounded degree graphs, for any graph $\mathscr{G}$, radius r and $\varepsilon > 0$ there always exists a graph $\mathscr{H}$ whose size is independent of $|V(\mathscr{G})|$ and whose r-neighbourhood distribution vector satisfies $\|\operatorname{dv}_r(\mathscr{G}) - \operatorname{dv}_r(\mathscr{H})\|_1 \leq \varepsilon$. This easily extends to relational databases. The proof of this result is only existential, however, and does not provide explicit bounds on the size of $\mathscr{H}$, it was suggested as an open problem by Alon to find such bounds. In Chapter 4 we found explicit bounds on the size of $\mathscr{H}$ for graphs/databases from a class of graphs/databases whose histogram vectors form a semilinear set, and for hyperfinite graphs/databases.

7.1.3 Approximate enumeration

In Chapter 6 we introduced a new model for the problem of *approximate FO query enumeration* on relational databases of bounded degree. For a query q, class of bounded degree databases $\mathbf{C}$ and database $\mathscr{D} \in \mathbf{C}$, we relax the set of answers $q(\mathscr{D})$ by admitting tuples from the superset $q(\mathscr{D}, \mathbf{C}, \varepsilon) \supseteq q(\mathscr{D})$, a set of ε-*approximate* answers, that are structurally close to being in $q(\mathscr{D})$ (note that for local FO queries $q(\mathscr{D}, \mathbf{C}, \varepsilon) = q(\mathscr{D})$). Our enumeration algorithms, with high probability (1) only enumerate tuples in $q(\mathscr{D}, \mathbf{C}, \varepsilon)$, and (2) if $|q(\mathscr{D})| \geq f(n)$, where $f(n)$ is the answer threshold function, then every tuple in $q(\mathscr{D})$ is enumerated.

We proved that for any local FO queries q with arity k and $\gamma \in (0,1)$, on any class of databases $\mathbf{C}$ of bounded degree, the enumeration problem $\text{Enum}_{\mathbf{C}}(q)$ can be solved approximately with a constant time preprocessing phase and constant delay for answer threshold function $f(n) = \gamma n^k$ (Theorem 6.13). Furthermore, for local FO queries with probability 1 we only enumerate actual answers. We then proved that in our model, for all general FO queries q with arity k and $\gamma \in (0,1)$, the enumeration problem $\text{Enum}_{\mathbf{C}_d^t}(q)$

can be solved approximately with a preprocessing phase that runs in polylogarithmic time and constant delay for answer threshold function $f(n) = \gamma n^k$ (Theorem 6.19). We actually strengthened both of these results by reducing the answer threshold functions to $f(n) = \gamma n^c$ (Theorems 6.14 and 6.20), where c is the maximum number of connected components in the (fixed radius) neighbourhood of an answer tuple to q for any database. In practice we would expect c to be small.

We generalised our result on the approximate enumeration of general FO queries. We identified the condition of Hanf-sentence testability (which is a weakening of the bounded tree-width condition), under which we still get the same probabilistic guarantees of Theorem 6.20.

Furthermore, we extended our model and results to the computational problems of *approximate query membership testing* (a relaxation of the exact query membership testing problem) and *approximate counting*. We showed that for every FO query q on the class of all bounded degree and bounded tree-width databases, (1) the problem of approximate query membership testing can be solved with a polylogarithmic preprocessing phase and constant time answering phase, and (2) we can estimate the number of tuples that are or are close to being answers in polylogarithmic time.

A natural question is, can the running time of the preprocessing phase of our enumeration algorithm for the problem $\mathrm{Enum}_{\mathbf{C}'_d}(q)$ (where q is any FO query) be improved to constant time? We answered this positively if we adopt the distance measure used in the $\mathrm{BDRD}_{+/-}$ model (rather than the distance measure used in the BDRD model) in our definition of close answers. It remains open whether the preprocessing phase running time can be improved to constant whilst using the distance measure of the BDRD model.

7.2 Future work

7.2.1 Approximate boolean query evaluation

Obtaining a characterisation of constant query testable properties is a long-standing open problem in the bounded degree graph model. It would also be interesting to obtain a characterisation of constant query testable properties in the $\mathrm{BDRD}_{+/-}$ model, or better yet obtain a characterisation of efficiently (sublinear running time) constant query testable properties. In the bounded degree graph model Ito, Khoury and Newman [48] give a characterisation of the 1-sided error constant query testable monotone and hereditary graph properties. A starting point in obtaining a characterisation of constant query testable properties in the $\mathrm{BDRD}_{+/-}$ model would be to see if the characterisation given in [48] carries over to the $\mathrm{BDRD}_{+/-}$

model. One of the main tools used in [48] is that every testable property has a *canonical tester*. A canonical tester is a testing algorithm that samples vertices uniformly at random, explores the r-neighbourhood (for some r) around the sampled vertices, and then makes its decision based on the explored neighbourhoods. To translate the results of [48] a crucial step would be to prove that in the $\text{BDRD}_{+/-}$ model every testable property has a canonical tester.

In Chapter 5 we showed that every property that is uniformly testable in the BDRD model is uniformly testable in the $\text{BDRD}_{+/-}$ model but the converse is not true. We gave an example of a property that is (trivially) uniformly testable in the $\text{BDRD}_{+/-}$ model but is not uniformly testable in the BDRD model. It would be interesting to obtain other such examples of properties that are uniformly testable but not necessarily trivially uniformly testable in the $\text{BDRD}_{+/-}$ model. It would also be interesting to find properties that are testable in the BDRD model but have more efficient testers in the $\text{BDRD}_{+/-}$ model. For example, can it be shown that properties definable by a CMSO sentence on databases of bounded degree and bounded tree-width cannot be tested in constant time in the BDRD model?

In Chapter 5 we showed that being hyperfinite and semilinear (the histogram vectors of the databases in the property form a semilinear set), or even just close to semilinear, gives us constant time uniform testability in the $\text{BDRD}_{+/-}$ model. Properties definable by a CMSO sentence on databases of bounded degree and bounded tree-width are examples of such properties. We also proved that hyperfinite hereditary properties are close to being semilinear. In the future, it would be good to obtain other examples of hyperfinite properties that are close to being semilinear. Furthermore, it would be interesting to try and understand the exact link between semilinearity and efficient uniform testability in the $\text{BDRD}_{+/-}$ model.

7.2.2 Approximate enumeration

In Chapter 6 we only considered bounded degree databases. It would be interesting to explore what results we can obtain if we consider more general classes of sparse databases, for example, databases with bounded average degree or low degree. We would need to use a different property testing model, and it would require different techniques because the number of r-types up to isomorphism is no longer bounded and we can no longer compute the r-type of a tuple in constant time. In our enumeration algorithms we only considered queries definable in FO. We could consider other query languages such as CMSO.

In our approximate enumeration algorithms we require the set of answers to a query to be sufficiently large in order to output every answer with high probability. We could investigate how to reduce this required answer set size. To do this we would need new ideas and we may possibly need a new framework.

In this thesis we have only considered enumeration in the static setting. There has been numerous results on the exact enumeration of queries in the dynamic setting (where databases may be updated by inserting or removing tuples). It would be interesting to study approximate enumeration in the dynamic setting.

References

[1] Isolde Adler and Polly Fahey. Faster property testers in a variation of the bounded degree model. Submitted for publication.

[2] Isolde Adler and Polly Fahey. Towards approximate query enumeration with sublinear preprocessing time. Submitted for publication.

[3] Isolde Adler and Polly Fahey. Faster property testers in a variation of the bounded degree model. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14–18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 7:1–7:15. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[4] Isolde Adler and Frederik Harwath. Property testing for bounded degree databases. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28–March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 6:1–6:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

[5] Noga Alon, W. Fernandez De La Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of MAX-CSPs. *Journal of computer and system sciences*, 67(2):212–243, 2003.

[6] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(4):451–476, 2000.

[7] Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009.

[8] Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics*, 22(2):786–819, 2008.

[9] Noga Alon, Paul D. Seymour, and Robin Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 293–299, New York, NY, USA, 1990. Association for Computing Machinery. doi:10.1145/100216.100254.

[10] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008.

[11] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley, 1992.

[12] Brian Babcock, Surajit Chaudhuri, and Gautam Das. Dynamic sample selection for approximate query processing. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, pages 539–550. Association for Computing Machinery, 2003.

[13] Jasine Babu, Areej Khoury, and Ilan Newman. Every property of outerplanar graphs is testable. In José D. P. Rolim Klaus Jansen, Claire Mathieu and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7–9, 2016, Paris, France*, pages 21:1–21:19. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2016.

[14] Guillaume Bagan. MSO queries on tree decomposable structures are computable with linear delay. In Zoltán Ésik, editor, *International Workshop on Computer Science Logic*, pages 167–181. Springer, 2006.

[15] Guillaume Bagan, Arnaud Durand, Etienne Grandjean, and Frédéric Olive. Computing the jth solution of a first-order query. *RAIRO-Theoretical Informatics and Applications*, 42(1):147–164, 2008.

[16] Sagi Ben-Moshe, Yaron Kanza, Eldar Fischer, Arie Matsliah, Mani Fischer, and Carl Staelin. Detecting and exploiting near-sortedness for efficient relational query evaluation. In Tova Milo, editor, *Proceedings of the 14th International Conference on Database Theory, Uppsala, Sweden, March 21–23, 2011*, pages 256–267. ACM, 2011.

[17] Itai Benjamini, Oded Schramm, and Asaf Shapira. Every minor-closed property of sparse graphs is testable. *Advances in mathematics*, 223(6):2200–2218, 2010.

[18] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. Answering FO+ MOD queries under updates on bounded degree databases. *ACM Transactions on Database Systems (TODS)*, 43(2):1–32, 2018.

[19] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998. URL: https://www.sciencedirect.com/science/article/pii/S0304397597002284, `doi:https://doi.org/10.1016/S0304-3975(97)00228-4`.

[20] Benedikt Bollig and Dietrich Kuske. An optimal construction of Hanf sentences. *Journal of Applied Logic*, 10(2):179–186, 2012.

[21] Edouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 601–612, 2020.

[22] Kaushik Chakrabarti, Minos Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Approximate query processing using wavelets. *The VLDB Journal*, 10(2):199–223, 2001.

[23] Hubie Chen, Matt Valeriote, and Yuichi Yoshida. Constant-query testability of assignments to constraint satisfaction problems. *SIAM Journal on Computing*, 48(3):1022–1045, 2019.

[24] Hubie Chen and Yuichi Yoshida. Testability of homomorphism inadmissibility: Property testing meets database theory. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 365–382. ACM, 2019.

[25] Bruno Courcelle. Chapter 5 - graph rewriting: An algebraic and logic approach. In Jan Van Leeuwen, editor, *Formal Models and Semantics*, Handbook of Theoretical Computer Science, pages 193–242. Elsevier, Amsterdam, 1990.

[26] Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2012.

[27] Artur Czumaj, Asaf Shapira, and Christian Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6):2499–2510, 2009.

[28] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 270–279. IEEE, 2007.

[29] Arnaud Durand and Etienne Grandjean. First-order queries on structures of bounded degree are computable with constant delay. *ACM Transactions on Computational Logic (TOCL)*, 8(4):21–es, 2007.

[30] Arnaud Durand, Nicole Schweikardt, and Luc Segoufin. Enumerating answers to first-order queries over databases of low degree. 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 121–131. ACM, 2014. doi:10.1145/2594538.2594539.

[31] Zdeněk Dvořák, Daniel Král, and Robin Thomas. Deciding first-order properties for sparse graphs. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 133–142. IEEE, 2010.

[32] Hendrik Fichtenberger, Pan Peng, and Christian Sohler. On Constant-Size Graphs That Preserve the Local Structure of High-Girth Graphs. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 786–799, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[33] Hendrik Fichtenberger, Pan Peng, and Christian Sohler. Every testable (infinite) property of bounded-degree graphs contains an infinite hyperfinite subproperty. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 714–726. SIAM, 2019.

[34] Eldar Fischer and Johann A Makowsky. On spectra of sentences of monadic second order logic with counting. *Journal of Symbolic Logic*, 69(3):617–640, 2004.

[35] Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM (JACM)*, 48(6):1184–1206, 2001.

[36] Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1):3–31, 2004.

[37] Haim Gaifman. On local and non-local properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105–135. Elsevier, 1982.

[38] Venkatesh Ganti, Mong-Li Lee, and Raghu Ramakrishnan. ICICLES: Self-tuning samples for approximate query answering. In *Proceedings of the 26th International Conference on Very Large Data Bases*, page 176–187, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[39] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

[40] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

[41] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 406–415, 1997.

[42] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

[43] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM (JACM)*, 64(3):1–32, 2017.

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

[45] Avinatan Hassidim, Jonathan A. Kelner, Huy N. Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 22–31, 2009. doi:10.1109/FOCS.2009.77.

[46] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 171–182, 1997.

[47] Piotr Indyk, Andrew McGregor, Ilan Newman, and Krzysztof Onak. Open problems in data streams, property testing, and related topics, 2011. *Available at: people.cs.umass.edu/mcgregor/papers/11-openproblems.pdf*, 2011.

[48] Hiro Ito, Areej Khoury, and Ilan Newman. On the characterization of 1-sided error strongly testable graph properties for bounded-degree graphs. *Computational Complexity*, 29(1):1–45, 2020.

[49] Wojciech Kazana. *Query evaluation with constant delay*. PhD thesis, École normale supérieure de Cachan, Paris, France, 2013.

[50] Wojciech Kazana and Luc Segoufin. First-order query evaluation on structures of bounded degree. *Logical Methods in Computer Science*, 7(2), 2011. doi:10.2168/LMCS-7(2:20)2011.

[51] Wojciech Kazana and Luc Segoufin. Enumeration of first-order queries on classes of structures with bounded expansion. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '13, page 297–308, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2463664.2463667.

[52] Wojciech Kazana and Luc Segoufin. Enumeration of monadic second-order queries on trees. *ACM Transactions on Computational Logic (TOCL)*, 14(4):1–12, 2013.

[53] Phokion G Kolaitis and Moshe Y Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[54] Leonid Libkin. *Elements of finite model theory*. Springer, 2004.

[55] László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[56] Bernard M. E. Moret and Henry D. Shapiro. *Algorithms from P to NP (Vol. 1): Design and Efficiency*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1991.

[57] Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013.

[58] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[59] Nicole Schweikardt, Luc Segoufin, and Alexandre Vigny. Enumeration for FO queries over nowhere dense graphs. In Jan Van den Bussche and Marcelo Arenas, editors, *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 151–163. ACM, 2018. doi:10.1145/3196959.3196971.

[60] Detlef Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

[61] Luc Segoufin and Alexandre Vigny. Constant delay enumeration for FO queries over databases with local bounded expansion. In Michael Benedikt and Giorgio Orsi, editors, *20th International Conference on Database Theory (ICDT 2017)*, pages 20:1–20:16, 2017.

[62] Saravanan Thirumuruganathan, Shohedul Hasan, Nick Koudas, and Gautam Das. Approximate query processing using deep generative models. *arXiv preprint arXiv:1903.10000*, 2019.

[63] Hanghang Tong, Christos Faloutsos, Christos Faloutsos, Brian Gallagher, and Tina Eliassi-Rad. Fast best-effort pattern matching in large attributed graphs. In Xindong Wu Pavel Berkhin, Rich Caruana and Scott Gaffney, editors, *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746. ACM, 2007.

[64] Alexandre Vigny. Dynamic query evaluation over structures with low degree. *arXiv preprint arXiv:2010.02982*, 2020.

[65] Yuichi Yoshida. Optimal constant-time approximation algorithms and (unconditional) inapproximability results for every bounded-degree CSP. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2011.

[66] Shijie Zhang, Shirong Li, and Jiong Yang. GADDI: distance index based subgraph matching in biological networks. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 192–203. ACM, 2009.