Lower bounds and structural results in property testing of dense combinatorial structures

Eyal Rozenberg

Lower bounds and structural results in property testing of dense combinatorial structures

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Eyal Rozenberg

Submitted to the Senate
of the Technion — Israel Institute of Technology
Tevet 5772 Haifa January 2012

This research was carried out under the supervision of Prof. Eldar Fischer, in the Faculty of Computer Science.

Most results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

Eldar Fischer and Eyal Rozenberg. Lower bounds for testing forbidden induced substructures in bipartite-graph-like combinatorial objects. In *Proceedings of RANDOM 2007*, pages 464–478. Springer, 2007.

Eldar Fischer and Eyal Rozenberg. Inflatable graph properties and natural property tests. In *Proceedings of RANDOM 2011*, pages 542–554, Berlin, Heidelberg, 2011. Springer-Verlag.

Oded Goldreich, Michael Krivelevich, Ilan Newman, and Eyal Rozenberg. Hierarchy theorems for property testing. In Oded Goldreich, editor, *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 289–294. Springer, 2010.

The Technion's funding of this research is hereby acknowledged.

Contents

Abstract

This thesis endeavors to deepen the understanding of the dense model for testing properties of combinatorial structures such as graphs, hypergraphs, matrices and tensors. This is achieved through the development of structural concepts regarding testing in the dense model, which are then put to use: In formulating new lower bounds on the query complexity for testing certain classes of such properties; in enhancing known lower bounds; and in achieving hierarchy results with both upper and lower bounds.

We first focus on dense graphs, and consider *natural* testing: Property tests which act entirely independently of the size of the graph being tested. We introduce the notion of graph properties being *inflatable* — closed under taking (balanced) blowups — and show that the query complexity of natural tests are related to the degree to which a property is approximately hereditary and approximately inflatable. Specifically, we show that for properties which are almost hereditary and almost inflatable, any test can be made natural with little penalty in the number of queries. In the reverse direction, we show that properties admitting natural tests are approximately inflatable and approximately hereditary, with parameters depending on the test's number of queries. Using the technique for naturalization, we restore in part a claim of Goldreich and Trevisan regarding testing hereditary properties, and generalize the relation between one-sided and two-sided lower bounds on triangle-freeness testing; we also give a simple explicit proof of a slight improvement of the best current explicitly-stated lower bound on triangle-freeness testing. More generally, we explore the relations of the notion of inflatability and other already-studied features of properties and property tests in the dense graph model, such as one-sidedness, heredity, and proximity-oblivion. Finally, we generalize these results to dense structures other than graphs.

From natural testing we turn to study tests which are highly-dependent on the size of their input graph: We construct a property of dense graphs which is maximally-hard to test, in terms of the number of queries necessary, but which can be efficiently decided, and whose test is time-efficient. Using this and some already-established constructions we prove several hierarchy theorems for the dense graph model, establishing that for every possible reasonable function of the input graph size, there exists properties with exactly this function as its query complexity — and with certain desirable features. We prove a similar hierarchy theorem both for testing generic functions and graphs in the sparse testing model. As with the results regarding natural tests, in reaching these

results we further explore, and make extensive use of, the concept of graph blowups.

We next present several results regarding testing dense structures which are essentially different than (the more widely-studied) general graphs.

We give lower bound results regarding testing bipartite graphs with colored edges, and k-partite k-uniform hypergraphs (which can be seen as testing matrices and tensors over fixed finite fields, if coordinate order is disregarded). In this context, a previous positive result showing that bipartite graphs are easily testable for freeness of forbidden induced subgraphs is shown not to hold when edges can have multiple colors, or when the 'dimension' is increased to k-partite k-uniform hypergraphs with $k \geq 3$. A lower bound is obtained, settling an open question of Alon, Fischer and Newman.

Two final results regard testing properties of general hypergraphs with multiple edge relations (or colors), and more specifically, properties which are characterized by partitions of vertex tuples, with density constraints on these partitions. We show that such properties can be efficiently 'pseudo-tested', that is, one can distinguish whether or not there exist partitions which approximately satisfy the density constraints. However, this 'pseudo-testing', sufficient for obtaining an actual test for partition properties of graphs, or of partitions of hypergraph vertices only, does not suffice in the general case — as we are able to demonstrate by proving a lower bound on the query complexity of such hypergraph properties.

These results are based, for the most part, on articles published by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being [FR07], [GKNR10] and [FR11].

Chapter 1

Introduction

Studies in Theoretical Computer Science, and specifically in Computational Complexity, are most often concerned with the following kind of question: How much of a certain computational resource is necessary, or sufficient, for solving certain computational problems? The resource of interest can typically be the computation time, or the number of operations: How fast can one, say, sort an array of numbers, as a function of its length? Other resources often studied are the amount of memory space for performing the computation, or the number of bits of communication necessary for several computers interacting over a network to compute something in collaboration. The field of *Property Testing* can be thought of as the study of how much *information from the input instance* of a computational problem is necessary for making a valid decision.

Of course, one can generally not make correct decisions with certainty about an input object — a string of characters, a graph, a function evaluated over a certain domain — without reading it in its entirety; but one can very often reach certain conclusions about the entire input based on samples from it, with high probability of their being correct. More specifically, a *property test* is allowed oracle access to some combinatorial object, and must distinguish with high probability between the case of this object satisfying a certain property, and the case of the object being far from satisfying it by some measure of distance. Roughly, when one needs to change at least an ε-fraction of the representation of the object to make it satisfy the property, it is considered to be *ε-far* from satisfying the property. One is interested in devising tests making as few queries as possible of function values, presence of graph edges, matrix cell values, etc. A test, therefore, must probabilistically decide the promise problem, in which the input is guaranteed to either satisfy a property or be far from satisfying it; and it is allowed to err or fail for inputs which are close to satisfying the property, but do not quite satisfy it.

Such problems were first studied by Blum, Luby and Rubinfeld in [BLR90], which was concerned with testing the linearity of functions, and began a long line of inquiry into testing algebraic properties; one of these works, by Rubinfeld and Sudan [RS96],

first introduced the general formulation of Property Testing as such.

The study of testing properties of *combinatorial* objects began with the work of Goldreich, Goldwasser and Ron in [GGR98], regarding properties of graphs. Combinatorial property testing has been an active field of research in the decade-and-a-half since, as is evidenced by the earlier surveys by Fischer [Fis04] and by Ron [Ron01], and the more recent survey of by Ron [Ron10] and by Goldreich [Gol10] (the latter is in fact a collection of mini-surveys and articles, including [GKNR10]).

Dense structure testing and other models

One of the important aspects in the study of property testing is the *testing model* — that is, exactly what information is the test given in advance; what is the distance metric between input structures; and what kind of queries it can make regarding the implicit input structure (or, alternatively, how is the input structure represented). For example, in the case of graphs, the test might ask "is there an edge between the i^{th} and j^{th} vertices?" or it might ask "which vertex is the k^{th} neighbor of the i^{th} vertex?" — with these kinds of queries corresponding to an adjacency-matrix representation of a graph or an adjacency-list representation, respectively.

The testing model corresponding to an adjacency-matrix representation of a graph is called the *dense model*. This was the first model considered for testing graph properties, introduced in [GGR98]. In this model, graphs on n vertices are ε-close to each other if one needs to add and/or remove an ε-fraction of all possible $\binom{n}{2}$ edges from one graph to convert it into the other — an ε-fraction of the representation of the graph. As the properties concern graphs rather than representations (in which vertices are labeled), the set of representations of satisfying graphs in the model must be closed under graph isomorphism, so if a certain labeled graph is considered to satisfy the property, so are all labeled graphs obtained from it by permuting the labels. In the dense model, sparse graphs (with $o(n^2)$ edges) are all close to being empty by this definition, hence the model's name.

A second model which has been the focus of research is the *bounded-degree model*, corresponding to an adjacency-list representation of graphs. In this model, introduced by Goldreich and Ron in [GR02], each vertex' degree is bounded by a fixed value d, and the test can query a vertex to obtain any of its up to d neighbors. Asymptotically, as $d \ll n$, such graphs are all so sparse that in the dense model they would be indistinguishable from the empty graph, and could be safely treated as empty. In the sparse graph model also, the distance is the fraction of the total possible edges necessary to convert one graph into the other — but in sparse graphs, a number of edges linear in n suffices to make two graphs far from each other.

This difference between the models is not merely 'fine' versus 'coarse' resolution; specifically, a property may be non-trivial (and not-maximally-hard) to test, in both of these models. A telling example is the property of *bipartiteness* — the vertex set

being divisible into two subsets, with no edges within each set. In the dense model, the complexity of bipartiteness is $\tilde{\Omega}\left(\varepsilon^{-1.5}\right)$ (due to Bogdanov and Trevisan in [BT04]) and $O\left(\varepsilon^{-2}\right)$ (due to Alon and Krivelevich in [AK99]); in the bounded-degree model the complexity is $\tilde{\Omega}(\sqrt{n})$ (presented with the introduction of the model, in [GR02]) and $\tilde{O}(\sqrt{n} \cdot \mathrm{poly}(1/\varepsilon))$ (in the subsequent [GR99] by the same authors).

These models do not cover the entire possible range of graph densities, and indeed, Krivelevich, Kaufman and Ron have considered a model 'mixing' the queries possible in the dense and the sparse models, in [KKR04] (exploring bipartiteness for different graph densities from sparse to dense) and [AKKR08] with Alon, as well as a graph testing model with stronger queries in [BEKKR10] with Ben-Eliezer.

This thesis focuses on testing in the dense model. However, dense testing is not limited merely to graphs, and extends readily to other kinds of structures: A structure's representation includes a set or several sets of vertices, as well as a fixed number of relations (collections of tuples), or collection of sets, with limited arity or set size. One can thus consider the testing of dense digraphs, partite graphs, graphs with edge colors, matrices and tensors, or more generally — hypergraphs, with or without edge orientation, with one or more edge relations (or with edge 'colors'). The 'denseness' carries to different structures through the normalized Hamming distance metric: An ε-fraction of modifications out of the total number of possible edges/tuples/sets, or number of matrix/tensor cells etc., makes two structures far from each other, and sparse structures are regarded as virtually-empty.

As in the case of graphs, properties must be closed under permutations of the vertices, so that any labeling or ordering of vertices in the representation of the structure do not carry information; if one is testing binary matrices, for example, the two matrices $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ represent the same object and both satisfy or fail to satisfy a given property.

The above example immediately leads one to consider another extension of the dense model, to structures such as ordered matrices and hypergraphs with vertex order. While some of the research work leading to this thesis concerned such structures, they have thus far failed to produce any results of note, and they are therefore not explored in this thesis. However, Fischer and Newman's [FN07a] studies some specific properties of multi-dimensional tensors with a partial order on their cells.

Testable and hard-to-test graph properties in the dense model

One wishes to be able to characterize which properties admit which kinds of tests: What dependencies can one achieve of the necessary number of queries on n and ε, and what useful features can tests be shown to have. Interestingly, [GGR98] demonstrated that certain (graph) properties, such as k-colorability, while being **NPTIME**-hard as decision problems, admit very efficient tests in the dense model — using a number of queries independent of the size of the input graph, and depending only on the distance

parameter ε; such properties are referred to as being *testable*. [GGR98] established a large class of properties as testable, and posed the characterization of the class of properties testable in the dense model as an open problem. In the following decade, a series of results gradually progressed towards this goal, and a characterization was achieved in Alon, Fischer and Newman's [AFNS09], and independently by Borgs, Chayes, Lovász, Sós, Szegedy and Vesztergombi in [BCL$^+$06] (in terms of graph limits).

The main technical tool in these works is Szemerédi 's *regularity lemma*, stating that large enough graphs can be decomposed into a bounded number of bipartite graphs most of which are similar to random graphs (see Szemerédi 's own [Sze78] for the original lemma, Fischer's [Fis04, Section 5] for basic discussion of its use for testing, or the characterization result in [AFNS09] itself). Unfortunately, using it incurs a prohibitive dependence on ε — while many significant properties have a mere polynomial dependence on ε in the number of queries. Thus the question of the dependence of the query complexity on ε has remained a significant avenue of research.

On the other end of the spectrum from testable properties are those properties whose query complexity is 'maximally' dependent on n — with query complexity $\Theta(n^2)$; artificial such properties were presented already in [GGR98]. Between the extremes, certain properties have been established to have various query complexity functions, (e.g. constant powers of n below 2, as in [FM06, PRR03]).

Relating features of properties, features of tests and query complexity

Within a given testing model, general results are often derived by further qualifying the model with certain features and obtaining bounds on query complexity or other provable consequences. These qualifications are usually features either of the property itself, or of the test. For example, a notable result on the way to characterizing the class of testable graph properties in the dense model is Alon and Shapira's [AS08a]: This work showed that if a property is hereditary, then it is also testable (that is, it admits a test whose number of queries is independent of the size of the input); a *hereditary* property is such that any induced subgraph of a satisfying graph is also itself a satisfying graph. In fact, it was established that hereditary properties are not only testable, but have tests with *one-sided error* (that is, tests that can never reject inputs satisfying the property, regardless of which queries they make).

Another example is of strengthening an existing upper or lower bound result on query complexity by making additional constraints on the property, as in Goldreich and Trevisan's [GT03, Theorem 1]: In this improvement of a result in [GGR98], the existence is demonstrated of properties requiring $\Omega(n^2)$ queries, which are not only in **NPTIME**, as was previously known, but also monotone; a graph property is *monotone (increasing)* if it is closed to adding edges, i.e. adding edges (but not vertices) to a satisfying graph results in another satisfying graph.

A third example regards the characterization of the 'power' of features of tests. Such

is a result of Goldreich and Ron in [GR10] (following the earlier work of Gonen and Ron in [GR07]) regarding adaptive tests; a test is *adaptive* if it considers results of previous queries when deciding which query to make next. [GR10] finds some testable graph properties in the dense model that exhibit a polynomial gap between an upper bound on the query complexity of adaptive tests, and a lower bound on the query complexity of non-adaptive tests. In the sparse graph model this gap can be exponential (assuming the test does not have to provide the labels of queried vertices in advance).

A more restricting feature of a test than being non-adaptive is being *canonical*, introduced in another section of [GT03]: A canonical test samples a number of vertices, and queries their entire induced subgraph; it then makes a deterministic decision whether to accept or reject the graph based on this small subgraph.

This thesis will present several results of a nature similar to these examples, as well as introduce certain hereto-unexplored features of properties of dense graph (and other dense structures).

Testing triangle-freeness

Perhaps the most studied class of properties in the dense model is that of being free of certain families of forbidden substructures, and specifically the property of being triangle-free. This property easily springs to mind once one begins to think up simple properties of graphs: A first non-trivial such property may be "not having edges", distinguishing empty graphs from graphs with many edges; after edges, perhaps paths, and then, perhaps a small cycle, a triangle. And while the query complexity obeing free of edges or of paths of any fixed length is easy to analyze (the query complexity is $\Omega(1/\varepsilon)$ queries), studying triangle-freeness testing is a very challenging endeavor: While the property is known to be testable, there is a vast gap between the lower and upper bounds for it.

The best known upper bounds for testing a graph for being free of triangles were until recently based on applying Szemerédi 's regularity lemma: See [Alo99], a proof sketch in [Fis04], or a more general treatment covering any family of induced subgraphs in Alon, Fischer, Krivelevich and Szegedy's [AFKS00]. This construction yields a query complexity equal to a tower function of height polynomial in $1/\varepsilon$ (even a double-tower for general forbidden induced subgraphs); recently, Fox has proven in [Fox11] a tower function upper bound for forbidden subgraphs, whose height is only logarithmic in $1/\varepsilon$, by a technique similar to the one used for proving Szemerédi 's Regularity Lemma, customized to the problem of subgraph-freeness.

The study of the property of triangle-freeness has also seen much use of the relations between features of properties and features of tests, for obtaining lower bounds. The standard approach for proving lower bounds on a property's query complexity is Yao's method, named after a principle observed in Yao's [Yao77]: if any deterministic test can't distinguish well enough between two fixed probability distributions, one over satisfying

graphs and one over far graphs, then no probabilistic algorithm (which is a distribution over deterministic algorithms) can do so either, and a lower bound is established — usually for non-adaptive tests. If the test is adaptive, proving indistinguishability becomes more complex, as queries depending on the test's history of queries already made can much better distinguish between input distribution.

If we limit our attention to one-sided tests only, things become somewhat simpler: A test querying a subgraph which in itself contains no triangles would have to accept, as it is possible that there are no edges in the graph except the queried ones. A bound therefore requires only constructing a single graph (for every order n) which has very few triangles, but no small set of edges intersecting all of them. Indeed, such a construction by Alon in [Alo02] established a bound (mildly) super-polynomial in $1/\varepsilon$; this bound is based on a number-theoretic construction of Behrend in [Beh46] of dense sets of integers without any three-term arithmetic progression. A recent construction by Elkin in [Elk11] of larger arithmetic-progression-free sets allows for a slight improvement of the [Alo02] bound.

If one could convert such one-sided lower bounds into general, two-sided bounds, this could be a shortcut avoiding a complex adversarial Yao's-method construction. And indeed, [GT03] includes a proposition communicated by Noga Alon: Testable hereditary properties can be tested by merely ensuring that most small induced subgraphs themselves satisfy the property (with a mild increase in the number of queries). Consequently, if the property is both hereditary and one-sided, then any test should imply the existence of a one-sided test — and any bound on one-sided testing becomes a bound on testing in general. Unfortunately, it later turned out that this proposition only holds for tests which are 'natural': Tests acting independently of the size of the input graph. This qualification appears in the errata [GT05].

Alon and Shapira worked in [AS06] around the hurdle of not being able to generalize the one-sided triangle testing lower bound of [Alo02] to the two-sided setting, by proving the same quasi-polynomial lower bound for any triangle freeness test, directly, using Yao's method to obtain specific indistinguishable distributions. However, this method is limited to a specific kind of constructions, and may not necessarily apply to future one-sided lower bounds.

1.1 Overview of results

Inflatable properties and natural property tests

In Chapter 3 (based on [FR11]) we establish links between the query complexity of natural tests and the features of graph properties being inflatable and hereditary. Specifically, we show that for properties which are almost hereditary and almost inflatable, any test with query complexity independent of n can be made natural, with a polynomial increase in its number of queries. The naturalization is carried out as a sort of extension of the

canonicalization due to Goldreich and Trevisan in [GT03], so that natural canonical tests can be described as *strongly canonical*. In the reverse direction, we show that properties admitting natural tests are approximately inflatable and approximately hereditary, with these parameters depending on the test's number of queries.

Using the technique for naturalization, we restore in part the claim in [GT03] mentioned above, regarding testing hereditary properties by ensuring that a small random subgraph itself satisfies the tested property. This restoration allows us to make a generalization regarding lower bounds on triangle-freeness testing: Any (future) lower bound — not only the currently established quasi-polynomial one — on one-sided testing for triangle freeness holds essentially for two-sided testing as well. We later make use of this generalization in the lower bounds for testing partite dense structures, in Chapter 5 (see overview below). We also demonstrate the use of this generalization through an explicit statement and simple proof of the bound implicit in the constructions of [Elk11], constituting a slight improvement over the best established lower bound of [AS06].

Finally, we prove a characterization of those inflatable properties which admit a proximity-oblivious test.

Query complexity hierarchies for dense graphs and other models

In Chapter 4 (based on [GKNR10]) we consider the question of the existence of properties with arbitrary query complexity. We answer this question affirmatively, establishing hierarchies of query complexity classes for both the sparse and the dense model for graph testing. Loosely speaking, we prove that for every reasonable function $q(n)$, there exists a property of graphs which is not testable using $o(q(n))$ queries, but is testable using $O(q(n))$ queries.

For the sparse graph model, we establish the hierarchy theorem using a non-artificial, easy-to-formulate property for every $q(n)$: The property of being 3-colorable and having connected components of order at most $q(n)$. The $q(n)$-query test establishing the upper bound is one-sided.

For the dense model, we in fact prove three variant hierarchy theorems, each for some additional feature of the properties or the test:

- A hierarchy of query complexity classes of properties which are **PTIME**-decidable (as languages) and **PTIME**-testable — that is, properties with a test whose running time is polynomial in $q(n)$.
- A hierarchy for monotone properties (although not in **PTIME**).
- A hierarchy for properties in which the lower bound $q(n)$ on query complexity is matched by a one-sided upper bound, i.e. they can be one-sided tested with $q(n)$ queries.

Lower bounds for partite dense structures

In Chapter 5 (based on [FR07]) we consider dense structures other than general graphs: Bipartite graphs with colored edges and k-partite k-uniform hypergraphs — which correspond to matrices and tensors (with no order among rows and columns), binary or over finite domains. Relating to [AFN07], which established a polynomial upper bound for testing binary matrices for forbidden subgraph freeness, we prove super-polynomial lower bounds both for matrices over a trinary domain, and for 3-dimensional binary tensors; this shows that the upper bound result, and the concept of 'conditional regularity' underlying it, do not immediately extend to larger domains, nor to higher dimensions. The lower bound is based on a reduction from testing cycle-freeness in dense digraphs, utilizing also the result re-established in Chapter 3 regarding hereditary property lower bounds.

Pseudo-testing hypergraph tuple partition properties

In Chapter 6 we consider the prospects of expanding the set of efficiently-testable properties of hypergraphs with multiple (oriented) edge relations, as dense structures. Specifically, we consider a generalization of the graph partition properties established to be easily testable in [GGR98]. Fischer, Matsliah and Shapira show in [FMS07] that a rudimentary generalization of such partition properties to hypergraphs is also efficiently testable. We study a stronger and somewhat more expressive generalization, in which not only vertices are partitioned, but also vertex tuples of higher arity. We show that such a class of properties, while not being maximally expressive (e.g. it does not seem to allow expression of the property of having a regular hypergraph partition) does not have tests which are efficient in terms of ε. On the other hand, we show that they admit an efficient 'pseudo-test', which distinguishes hypergraphs satisfying such a property from hypergraphs for which every partition is far from being satisfactory; in other words, the pseudo-test may err for hypergraphs which are far from the property but have approximately-satisfying partitions. Unlike the case of graphs, having such an approximately-satisfying partition does not imply closeness to having a properly-satisfying one.

Chapter 2

Preliminaries

2.1 The dense model for property testing

This thesis concerns testing properties of dense combinatorial structures, with graphs being the most commonly studied, and for which the testing model is usually defined. As much of the thesis concerns other 'dense' structures (a concept which will be defined shortly), we first define the model for the case of graphs, and then make definitions for more general dense structures in Subsection 2.1.1.

In the context of this work, we refer to simple graphs, $G = (V, E)$, with V being a set of vertices of order n and E an edge set containing unordered pairs of vertices.

Definition 2.1.1. The *absolute distance* between two graphs G, H of order n is the number of edges one has to add and/or remove in G to make it into an isomorphic copy of H; in other words, it is the minimum over all bijections $\phi : V(G) \to V(H)$ of the number of edge discrepancies — the symmetric difference

$$\{\{u, v\} \in E(G) \mid \{\phi(u), \phi(v)\} \notin E(H)\} \cup \{\{u, v\} \in E(H) \mid \{\phi(u), \phi(v)\} \notin E(G)\}$$

The *(relative) distance* $\operatorname{dist}\big(G, H\big)$ between G and H is the absolute distance between them normalized by a factor of $\binom{n}{2}^{-1}$.

Two graphs are said to be *ε-far* if their distance is at least ε (that is, they have at least $\varepsilon\binom{n}{2}$ edge discrepancies).

Definition 2.1.2. A *property* of graphs is a set $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ of graphs, closed under graph isomorphism, where Π_n is supported on graphs of order n.

A graph of order n is said to *satisfy* a property Π if it is an element of Π_n; a graph is said to be *ε-far* from satisfying a property Π if it is ε-far from every graph $H \in \Pi_n$.

Definition 2.1.3. A *dense model property test* for a graph property Π is a probabilistic oracle machine which, given the values (n, ε), as well as oracle access to a graph G of

order n, makes a certain number of edge queries ("is there an edge between the vertices u and v?"), and distinguishes with probability at least $2/3$ between the case of G being in Π and the case of G being ε-far from Π. The (possibly adaptive) number and choice of queries, as well as the rest of the algorithm, may in general depend on the value of n, as can the decision to accept or reject.

Note. Many results regard tests for specific values of ε, rather than tests receiving ε as a parameter. Alon and Shapira prove in [AS08b] that these notions are different, with some properties only being testable with ε-specific tests rather than a general test receiving ε as a parameter. (The difference has to do with the computational tractability of the number of queries as a function of ε; see [Sha06, Chapter 3] for further discussion.) The results of this thesis hold for both settings. Specifically, all upper bounds are tests receiving ε as a parameter, while all lower bounds apply to ε-specific tests as well as tests receiving ε as a parameter.

Definition 2.1.3, the traditional definition of a property test in the dense model, includes an artificial dependence of the query model on the value of n: Without utilizing this value it is not possible to make any queries. The results and observations in [GT03, Section 4] emphasize the artifice of this particular dependence, and lead to an alternative definition of a test avoiding it:

Definition 2.1.4 (Alternative to Definition 2.1.3)**.** A *dense model property test* for a graph property Π is a probabilistic oracle machine which is given the values (n, ε), as well access to a graph G of order n, through an oracle which takes two types of requests: A request to uniformly sample an additional vertex out of the remaining vertices of G, and an edge query within the subgraph induced by the sampled vertices ("is there an edge between the i^{th} and j^{th} sampled vertices?"). The machine makes a sequence of requests to the oracle, and distinguishes with probability at least $2/3$ between the case of G being in Π and the case of G being ε-far from Π. If the test has sampled all vertices of the graph being tested, additional requests to sample an additional vertex will indicate that there are none left.

Definition 2.1.3 and Definition 2.1.4 are not equivalent as computational models in general, but in the context of testing dense structure properties closed under isomorphism — they are equivalent. With respect to graphs, this is established for all intents and purposes in [GT03], albeit not formally stated there. Further discussion of this point regarding dense structures in general can be found in Section 3.9.

2.1.1 General dense structures

A wide variety of dense structures are studied in this and other works on Property Testing, so that a "most-general definition" covering them all would make for a sort of a swiss-army-knife: General and partite graphs and hypergraphs; matrices and tensors,

over binary or other domains; hypergraphs with uniform-arity hyperedges or multiple arities; structures might have edges as sets of vertices, or as tuples as in the case of digraphs, or both; there may be a single edge relation, or multiple relations; et cetera. So as to state at least in a mostly-general way what constitutes a dense structure, we shall use the following:

Definition 2.1.5. An *unconstrained general dense structure* is a hypergraph $H = ((V_1, \ldots, V_k),\ (E_1, \ldots, E_t))$ with k vertex sets or parts, and t (hyper)edge relations (or "colors"), each being a set of arity-r_i tuples over the union of the vertex sets: $E_i \subseteq \prod_{i=1}^{r_i} \bigcup_{i=1}^{k} V_i$.

Definition 2.1.6. A *general dense structure class constraint* is a sentence in First-Order Logic without equality, with vocabulary $\{V_1, \ldots, V_k, E_1, \ldots, E_t\}$. The arity of each V_i symbol is 1, and the arity of each E_i is r_i. A constraint must have the form $\forall\, x_1 \ldots \forall\, x_s\ \varphi(x_1, \ldots, x_s)$, with φ being unquantified; the formula φ must be made up only of edge relation symbols of arity at least s, using all variables $x_1, \ldots, x_s$ at least once (but with possible repetitions), vertex part containment relation symbols (using a single variable), and Boolean connectives (including negation).

Definition 2.1.7. A *class of general dense structures* is the set of all unconstrained general dense structures with the same specific k, t and arities $(r_i)_{i=1}^{t}$, which satisfy a specific common set of constraints with the appropriate vocabulary, where the constraints are interpreted as follows: The domain is $\bigcup_i V_i$; the V_i symbols are interpreted as containing all vertices of the i^{th} part of the structure; and the E_i symbols are interpreted as the structure's own relations E_i.

Such constraints allow the expression of the wider variety of structures mentioned above through multi-relation hypergraphs. Some relevant examples:

- An edge relation may be constrained to be symmetric (permutation of the coordinates does not change the edge function value). An example: Structures with $k = 1$, $t = 1$ and $r_1 = 2$, with the constraint $\forall\, x\ \forall\, y\ \big[E_1(x, y) \leftrightarrow E_1(y, x)\big]$, are the expression of undirected graphs (with possible self-loops).

- Several edge relations (say, ℓ) of the same arity may be constrained to only have some of the 2^ℓ possible values for a certain tuple; this allows the expression of structures with colored edges, whose maximum number of colors is not a power of two, using multiple edge relations.

- A constraint may prevent tuples containing a single vertex more than once. For example, to prevent self-loops in graphs, the constraint imposed would be: $\forall\, x\ \big[\neg E_1(x, x)\big]$.

- An edge relation may be constrained to tuples in some specific sequence of vertex parts $V_{j_1} \times \ldots \times V_{j_{r_i}}$; this allows the expression of bipartite digraphs or k-partite

oriented hypergraphs. For bipartite digraphs (with $k = 2$, $t = 1$, $r_1 = 1$), the constraint would be: $\forall x \forall y \left[E_1(x_1, x_2) \leftarrow (V_1(x_1) \wedge V_2(x_1))\right]$.

Finally, while some structures do not technically fit even this general definition (e.g. matrices or tensors which have no vertices) — they can easily be expressed by a general dense structure with a simple transformation or reinterpretation (e.g. interpreting a matrix as the adjacency matrix of a bipartite graph). We will thus refer to them as dense structures as well.

Definition 2.1.8. A *property* of general dense structures of a certain class is a set Π of structures, all satisfying the constraints associated with the class, which is closed under isomorphism (i.e. closed under permutation of the vertices in each part).

Definition 2.1.9. For a class of general dense structures with one edge relation E of arity r, the *absolute distance* between structures in that class is defined as in the case of graphs, except that the discrepancies are between tuples rather than 2-sets. The *(relative) distance* is the absolute distance normalized by n^{-r}.

For classes with multiple edge relations, the absolute distance is not a meaningful concept, as the number of tuples in each edge relation is of a different order of magnitude with respect to to n. The (relative) distance, with respect to a specific bijection between the vertices of corresponding parts of the structures, is the maximum over all edge relations E_i of the number of discrepancies with respect to the bijection in that edge relation, normalized by n^{-r_i}. The overall *(relative) distance* is the the minimum of the above over all bijections.

Note. One can, as an alternative to the definition above, further normalize the distance by the maximum possible distance between two structures in the class (as in the case of simple undirected graphs, where the distance is a fraction of $\binom{n}{2}$).

Definition 2.1.10. A general dense structure with n vertices in each of its parts is said to be of *uniform order* n; if the number of vertices in each part differs, then the structure is said to be of (non-uniform) *order* $(n_1, n_2, \ldots, n_k)$.

Definition 2.1.11. A *dense model property test* for a property Π of a certain kind of dense structures is a probabilistic oracle machine which, given the values $(n_1, \ldots, n_k, \varepsilon)$, as well oracle access to a structure H with n_i vertices in each of the k parts , makes a certain number of tuple queries ("is the tuple $(x_1, \ldots, x_{r_i})$ in the edge relation E_i?"), and distinguishes with probability at least $2/3$ between the case of H being in Π and the case of H being ε-far from Π.

A *dense model uniform-order property test* for a property Π is a test as per the above, except that the structure tested is guaranteed to be of uniform order n, and the test is given the values (n, ε).

Note. The alternative definition for a property test in Definition 2.1.4, without an artificial dependence on the number of vertices (in each part), can be made similarly in the case of a general dense structure, with the oracle receiving either requests to sample a vertex from one of the parts of the graphs, or queries regarding the presence of tuples of already-sampled vertices in one of the structure's edge relations.

2.2 Features of dense structure property tests

As discussed in the introduction, it is interesting to distinguish tests not just by their use of computational resources (queries, running time, etc.) but also by other features specific to the setting of dense structure property testing or property testing in general.

Definition 2.2.1. A property test is said to be *one-sided* (or said to have *one-sided error*) if it accepts all graphs in Π with probability 1.

Definition 2.2.2. A property test is said to be *adaptive* if the queries it makes to the oracle may depend in some way on the results of previous queries. If no query made by the test depends on previous query results, the test is said to be *non-adaptive*.

Definition 2.2.3. A test for a graph property Π is said to be *canonical* if, for some function $s : \mathbb{N} \times (0,1) \to \mathbb{N}$ and some sequence of properties $\left(\Pi^{(i)}\right)_{i\in\mathbb{N}}$, the test operates as follows: On input n and oracle access to an n-vertex graph G, the test samples uniformly a set of $s(n,\varepsilon)$ distinct vertices of G, queries the entire corresponding induced subgraph, and accepts if and only if this subgraph is in $\Pi^{(n)}$. If the graph has fewer than $s(n,\varepsilon)$ vertices, the test queries the entire graph and accepts if it is in Π.

For a general dense structure, a canonical uniform-order test samples $s(n,\varepsilon)$ vertices from each one of the k parts, and queries the substructure induced by these sampled vertices. If the structure has fewer than $s(n,\varepsilon)$ vertices per part, the test queries the entire structure and accepts if it is in Π.

Note. For multi-partite dense structures, this definition is somewhat lacking — it does not cover tests of non-uniform-order structures. See Section 3.9 for further discussion and a reasoning for limiting the definition's scope in this work.

Definition 2.2.4 (as appearing in [GT05])**.** A (graph) property test is said to be *natural* if its query complexity is independent of the size of the tested structure, and on input (n,ε) and oracle access to a graph of order n, the test's output is based solely on the sequence of oracle answers it receives, and not on n (while possibly using more random bits, provided that their number and use is also independent of n).

If our graph property tests are as defined traditionally (Definition 2.1.3), the above definition of a natural test is flawed, and no test which makes any queries can be natural: A test cannot make $q(\varepsilon)$ queries to an input graph with less than $\sqrt{q(\varepsilon)}$ vertices (this

point is also mentioned in [AS08a]). Instead of amending the definition of naturality to avoid this semantic issue, it seems more reasonable to use the alternative definition for the dense graph model, Definition 2.1.4, in which the artificial dependence on n is removed. In this case, Definition 2.2.4 is valid: If the test attempts to sample too many vertices, the oracle indicates its failure to do so and the test proceeds accordingly. In fact, in this work the implicit assumption is made that whenever a test attempts to sample more vertices than the vertex set contains, the oracle indicates that this is the case, and the test proceeds to query the entire structure, accepting it deterministically if it satisfies the property being tested.

In Chapter 3 we further develop the notions of canonicality and naturality of tests, and explore their interrelation.

2.3 Features of dense structure properties

Definition 2.3.1. A property is said to be *testable* if it has a test whose maximum number of queries is independent of n, and depends only on ε. If the maximum number of queries is a polynomial function in $1/\varepsilon$, the property is said to be *polynomially testable*.

Definition 2.3.2. A graph property Π of is said to be *decidable in complexity class* **CLASS** if, for some reasonable string encoding of graphs (so that the string length is polynomial in the order of the graph), the language consisting of these encodings for the graphs of Π is in **CLASS**.

Thus a property is in **PTIME** if the language of Π graph encodings is accepted by a deterministic Turing machine running in time polynomial in the length of its input, etc. A similar definition can be made for non-graph structures — dense or otherwise.

Definition 2.3.3. A property of graphs is said to be *ε-testable in* **PTIME**, if it has an ε-test, whose running time is bounded by a polynomial function of its number of queries (rather than polynomial in n). The property is said to be *testable in* **PTIME** or **PTIME**-*testable* if it is ε-testable in **PTIME** for every $\varepsilon > 0$.

Definition 2.3.4. A property is said to be *hereditary* if it is closed under the taking of induced substructures.

Hereditary properties can be characterized by a (possibly infinite) set $\mathcal{F}$ of forbidden induced substructures — a structure satisfies a hereditary property Π if and only if it has no induced subgraph from the forbidden set $\mathcal{F}_\Pi$.

Definition 2.3.5. A property of graphs or hypergraphs is said to be *downwards monotone* if it is closed under the removal of edges (while maintaining the same number of vertices). If a property is closed under the addition of edges, it is said to be *upwards monotone*.

Note. This notion of monotonicity is wider than that used in [AS05], which combines the notions of monotonicity and heredity defined here.

Definition 2.3.6. A graph $G' = (V', E')$ is said to be a *blowup* of a graph $G = (V, E)$ if V' can be partitioned into $|V|$ clusters of vertices, each corresponding to a vertex in V, where the edges in E' between these clusters correspond to the edges of E. In other words, if $(u, v) \in E$ then the bipartite graph between the clusters corresponding to u and v is complete, and if $(u, v) \notin E$ then this bipartite graph is empty. G' must also have no edges within the clusters of such a partition. A graph blowup is said to be:

- an *exactly-balanced blowup* if the clusters in V' (corresponding to the vertices of G) all have exactly the same size (and, in particular, $|V|$ divides $|V'|$). In this case, for $t = |V'|/|V|$, G' is also said to be a *t-factor blowup* of G.
- a *balanced blowup* if all clusters are of size either $\lfloor |V'|/|V| \rfloor$ or $\lceil |V'|/|V| \rceil$. The unqualified term 'blowup' indicates a balanced blowup.
- a *generalized blowup* if all clusters in V' are non-empty (but have no other restriction on their sizes).
- a *relaxed generalized blowup* if the clusters in V' may have any size, with some possibly being empty.

The above definition requires an explicit statement for classes of general dense structures:

Definition 2.3.7. A dense structure $H' = ((V'_1, \ldots, V'_k), \ (E'_1, \ldots, E'_t))$ is said to be a *blowup* of a structure $H = ((V_1, \ldots, V_k), (E_1, \ldots, E_t))$ if it satisfies the following. First, each of its vertex sets V'_i can be partitioned into $|V_i|$ clusters of vertices, with each cluster C_v corresponding to some vertex $v \in V_i$. Additionally, the tuples in each E'_j correspond to the tuples in E_j: If $x = \left(x_1, \ldots, x_{r_j}\right) \in E_j$ then the complete r_j-uniform oriented hypergraph $\prod_{\ell=1}^{r_j} C_{x_\ell}$ is contained in E', and if $x \notin E$ then E'_j contains no hyperedge of this hypergraph. In particular, if H has no hyperedges involving the same vertex more than once, then H' has no hyperedges with more than one constituent vertex within the same cluster.

A blowup is said to be *balanced* if the clusters in each V'_i all have the same size up to a difference of at most 1; and *exactly-balanced* if the clusters have exactly the same sizes (and, in particular, $|V_i|$ divides $|V'_i|$). In this case, for $t_i = |V'|/|V|$, H' is also said to be a *$(t_1, \ldots, t_k)$-factor blowup* of H; if $t_i = t$ for all $i \in [k]$, the blowup is said to be a *t-factor blowup* of H.

Observation 2.3.8. General dense structure classes are, in themselves, closed to taking blowups: It is easy to verify that any constraint satisfied by a dense structure is also necessarily satisfied by a blowups of structures in the class. General dense structure classes are also closed to taking induced substructures, by a similar argument.

Definition 2.3.9. A property Π is said to be *inflatable* if it is closed under (balanced) blowups, i.e. if G satisfies Π, then so does any blowup of G.

The concept of inflatability, which this thesis introduces, is explored in Chapter 3.

2.4 Testing-Reductions between properties

The following definition is relevant essentially to any model for property testing, not merely dense graphs or other dense structures.

Definition 2.4.1. Consider two classes **CLASS** and **CLASS′** of combinatorial objects, each with some distance metric and some measure of object 'size' (e.g. the number of vertices in a graph or the number of bits in the representation of the object). Also, let $f : \mathbb{R}^+ \to [0,1]$ be a continuous function and $g, h : \mathbb{N} \to \mathbb{N}$. The testing of a property $\Pi \subseteq$ **CLASS**, in some testing model, is said to be (f_r, g_r, h_r)-*reducible* to the testing of property $\Pi' \subseteq$ **CLASS′** in another testing model if, given oracle access to a structure $K \in$ **CLASS** (with possible queries according to the testing model for Π), one may simulate an oracle to a structure $K' \in$ **CLASS′** (accepting queries according to the second testing model) with the oracle satisfying the following:

1. If K is of size n then K' is of size at most $O(h_r(n))$.
2. If $K \in \Pi$ then $K' \in \Pi'$.
3. If K is ε-far from Π (according to the **CLASS** metric) then K' is $f_r(\varepsilon)$-far from Π' (according to the **CLASS′** metric).
4. To answer a query regarding K', one must make at most $g_r(n)$ queries to K.

Abusing the definition somewhat, we shall sometimes describe Π as being *reducible to* Π'.

Lemma 2.4.2. *If, in the above settings, the query complexity of* Π' *is* $O(q(n,\varepsilon))$*, then the query complexity of* Π *is* $O(q(h_r(n), f_r(\varepsilon)) \cdot g_r(n))$*.*

Conversely, let $f(\varepsilon)$ *be continuous, with its image containing some interval* $(0, \varepsilon_0)$*, and let*

$$h_r{}^{-1}(n) = \min\{n' \in \mathbb{N} \mid h_r(n') = n\}$$
$$f_r{}^{-1}(\varepsilon) = \max\{\varepsilon' \in \mathbb{R}^+ \mid f_r(\varepsilon') = \varepsilon\}$$

With h_r *having an infinite image. If the query complexity of* Π *is* $\Omega\big(q'(n,\varepsilon)\big)$*, then the reducibility implies that the query complexity of* Π' *is* $\Omega\left(\frac{1}{g_r(h_r{}^{-1}(n))} \cdot q'\big(h_r{}^{-1}(n), f^{-1}(\varepsilon)\big)\right)$ *(for* $\varepsilon < \varepsilon_0$ *and the values of* n *for which* $h_r{}^{-1}$ *is defined).*

Proof. For the upper bound claim: Given a structure in **CLASS**, one applies the test for Π' (with $h_r(n)$ and $f_r(\varepsilon)$ instead of n, ε) while simulating oracle access to the

corresponding structure in **CLASS**$'$. This is by definition a valid test for Π', making the claimed number of queries.

For the lower bound one uses the reduction from testing Π to Π', obtaining a valid test as in the above. If the Π' test makes $o\Big(\frac{1}{g_r\left({h_r}^{-1}(n')\right)} \cdot q'\left({h_r}^{-1}(n'), f^{-1}(\varepsilon')\right)\Big)$ queries given n', ε', then when given $n' = h_r(n), \varepsilon' = f_r(\varepsilon)$, it makes $o\Big(\frac{1}{g_r(n'')} \cdot q'(n'', \varepsilon'')\Big)$ queries to the simulated oracle, for $n'' = {h_r}^{-1}(h_r(n)) \leq n$ and $\varepsilon'' = f^{-1}(f(\varepsilon)) \geq \varepsilon$, with each query requiring at most $g_r(n'')$ queries to the real oracle; thus actual number of queries is $o(q'(n'', \varepsilon'')) = o(q'(n, \varepsilon))$, contradicting the query complexity lower bound for Π. These last two steps of our argument can be made since the range of n'' is unbounded, and ε'' can be arbitrarily close to 0, so their $\limsup_{n'' \to \infty} n = \infty$ and $\liminf_{\varepsilon'' \to 0} \varepsilon = 0$. □

Observation 2.4.3. Reductions defined as per the above preserve one-sided error (in the construction of Π-testers from Π'-testers), but they do not necessarily preserve non-adaptivity if the query translation (item 4 above) is not itself non-adaptive.

Observation 2.4.4. If Π_1 is (f_1, g_1, h_1)-reducible to testing Π_2, and Π_2 is (f_2, g_2, h_2)-reducible to testing Π_3, then Π_1 is $(f_1 \circ f_2, g_1 \cdot g_2, h_1 \circ h_2)$-reducible to testing Π_3 — assuming that $h_2(n)$ is monotone increasing (otherwise one has to account more accurately for the $O(h_1(n))$ structure sizes resulting from the first reduction).

Chapter 3

Inflatable properties and natural property tests

3.1 Introduction

Goldreich and Trevisan's [GT03] includes two results tying several features of properties and tests together. Their article defined the feature of property tests being canonical (Definition 2.2.3). The article also demonstrates how any test can be made canonical with at most about a squaring of its number of queries; this immediately implies that the gap between properties' adaptive and non-adaptive query complexity (see the discussion in Chapter 1) is at most quadratic. The second result (due to Noga Alon) was mentioned in Chapter 1 with respect to triangle-freeness testing: If a property is hereditary, then a test for it can be replaced with merely ensuring that a small sampled subgraph satisfies the same property as the large one. However, the proof in [GT03] implicitly assumes that the test is natural (as in Definition 2.2.4); thus this result must be qualified, and is not usable as such for deriving lower bounds on testing a property in general.

It seems odd, however, that tests for hereditary properties could circumvent the argument in [GT03]. Many hereditary properties (specifically, those with finite families of forbidden graphs) are highly 'local' in their definition; wherefore might they benefit significantly from basing their action on the order of the entire input graph? If we constrain ourselves to properties with features preventing blatant 'pathologies' which preclude natural tests (e.g. the property of graphs having an odd number of vertices) — then one tends to believe that property tests are 'essentially natural', so that perhaps one can 'smooth out' any non-natural artificial dependence of tests on n.

The relevant features of properties allowing this adjustment will have to do with their heredity, on the one hand, and their inflatability on the other. For an intuition for the choice of these features, think of a property test as being canonical, with a set of acceptable subgraphs for each order n of the input graph; in general, this set may gain or lose elements as n increases or decreases; we want to 'fix' it somehow. Constraining a property to be hereditary intuitively 'covers' one direction of change in n : As the

input order increases, the set of forbidden subgraphs increasingly gains elements, so one expects the set of subgraphs accepted by the test to shrink gradually, or at least fail to grow. In the other direction, we would like the test's set of accepted subgraphs not to grow as n goes down; now, if whatever it is we accept at a certain order also appears at higher orders — through blowups — then we do not expect the set of accepted subgraphs to shrink. Again, this is merely intuition. A concrete immediate effect of requiring inflatability is precluding the pathology of graphs going from satisfying a property at order n to being very far from satisfying it by merely adding a vertex.

With regards to the idea of 'smoothing out' non-naturality, a typical example would be a test which arbitrarily rejects some specific queried subgraph at, say, even orders, and accepts it at odd ones. If this subgraph is very unlikely to appear in graphs in the property, then a natural test could be 'spoiled' by adding this behavior to it, while still remaining a valid test. However, this can only be done for a single possible queried subgraph, or few of them — such behavior is impossible with all acceptable graphs, or with any subset of them which has an overall high probability of being sampled. This leads one to recall that, in Alon, Fischer, Newman and Shapira's [AFNS09], the characterization of testability uses the set of all subgraphs of a fixed order accepted by a canonical test. Even more relevant is Fischer and Newman's [FN07b] (proving that testable properties are also estimable, a key result necessary for the characterization in [AFNS09]), where it is observed that if one has a good estimate of the subgraph distribution, then one knows in particular whether a test querying subgraphs of this order accepts with high probability or not. In fact, disregarding the heavy use of Szemerédi 's regularity lemma in [FN07b], its result is based mostly on estimating the subgraph distribution up to a small variation distance — an approach sometimes referred to as "meta-testing".

Indeed, by analyzing tests with a focus on the distribution of subgraphs of a fixed order and its behavior in subgraphs and blowups, under the constraints of heredity and inflatability (even with a little relaxation), tests can be made natural, with a polynomial penalty in the number of queries. This technique, the concept of inflatable properties, and some of the aspects of our analysis, allow us to achieve several related results — including a partial restoration of the proposition regarding testing hereditary properties — and to draw conclusions regarding lower bounds for testing triangle (and other induced subgraph) freeness.

3.2 Additional preliminaries

3.2.1 On features of properties and of tests

Canonicality

The definition of a property test being canonical appears above, as Definition 2.2.3. Any test can be made canonical:

Theorem ([GT03, Theorem 2]). *If a graph property has a test making $q(\varepsilon)$ queries involving at most $s(\varepsilon)$ vertices, independently of the size of the input graph, then it has a canonical test with queried subgraph order at most $9s(\varepsilon)$ (and query complexity $\mathrm{O}\big(q(\varepsilon)^2\big)$). If the original test is one sided, this canonical test's queried subgraph order is $s(\varepsilon)$ and it is also one-sided.*

Note. The theorem is not phrased in terms of the number of sampled vertices, but this is evident from the proof of theorem: The original test is repeated 9 times and the majority-vote is used, to amplify the probability of success from 1/3 to 1/6; see also [GT05, Page 2, Footnote 1]. If one wishes the canonical test to succeed with higher probability, this can be achieved by repeating the original pre-canonized test additional times (and using a majority vote) before applying canonization; the penalty is a constant-factor increase in the final order of the queried subgraph.

A canonical test, which accepts a graph G when the queried subgraph on its sampled vertices is G', is said to *accept G by sample G'*.

In this chapter we will be dealing mostly with tests which combine both the features of canonicality and naturality, focusing on making canonical tests natural as well. For canonical tests, the feature of naturality means that the 'internal' property, the one for which the sampled subgraph is checked for, does not depend on the order of the input graph. This observation leads us to use naturality to define several 'levels' of canonicality for a property test:

Definition 3.2.1. Consider a canonical test for graph property Π, with $\big(\Pi^{(i)}\big)_{i=1}^{\infty}$ being the sequence of properties the satisfaction of which the test checks for its sampled order-s subgraph. The test is said to be

perfectly canonical when $\Pi^{(n)} = \Pi$: The test does nothing but ensure that a small random subgraph satisfies the same property that the larger input graph is being tested for.

strongly canonical when $\Pi^{(n)} = \Pi'$: The test ensures that a small sampled subgraph satisfies some fixed property, the same one for any order of the input graph, but not necessarily Π itself.

weakly canonical for any $\big(\Pi^{(i)}\big)_{i=1}^{\infty}$: It may be the case that $\Pi^{(n)}$ is different for different input graph orders n.

Notes.

– Indeed, a test is strongly canonical if and only if it is both canonical and natural.
– In Alon and Shapira's [AS08a], the term *oblivious* is used for what we have defined as a strongly canonical test.
– There is only one perfectly canonical test for any queried subgraph order; of course, for many properties this will not constitute a test, as it will not distinguish satisfying graphs from far graphs with sufficient probability.

Approximate inflatability and heredity

We have defined what it means for a property to be inflatable and hereditary, in exact terms. In this chapter we require relaxations of these definitions, to be able to describe properties as approximately hereditary or approximately inflatable. These definitions will concern random subgraphs and "random blowups" of graphs, so we first discuss the latter briefly.

Definition 3.2.2. A *random blowup* of a graph from order n to order n' is the blowup in which the $n' \pmod n$ vertices have the larger clusters in the blowup (clusters of size $\lceil n'/n \rceil$ rather than $\lfloor n'/n \rfloor$) are chosen uniformly at random.

Definition 3.2.3. Let G, H be graphs of the same order, let $\pi : V(G) \to V(H)$ be a bijection achieving $\operatorname{dist}\big(G, H\big)$ (in terms of edge discrepancies), and let G' be a blowup of G. A blowup H' of H to the same order as G' is said to *correspond to* G' if for every $v \in V(G)$, the size of v's cluster in G' is the same as the size of $\pi(v)$'s cluster in H'. In other words, "the same" vertices in G and H get larger clusters.

Lemma 3.2.4. *Let $G \neq H$ be graphs of order n, let $n' > n$, and let $\pi : V(G) \to V(H)$ be a bijection achieving* $\operatorname{dist}\big(G, H\big)$*, i.e. exhibiting* $\operatorname{dist}\big(G, H\big) \cdot \binom{n}{2}$ *discrepancies. If one uniformly samples a blowup G' of G to order n', and applies a corresponding blowup to H, then the expected distance between the two blowups is strictly lower than* $\operatorname{dist}\big(G, H\big)$.

Proof. We show that the expected number of discrepancies under a bijection mapping each vertex v's cluster to a vertex in the cluster of $\pi(v)$ is less than $\operatorname{dist}\big(G, H\big)\binom{n'}{2}$, implying the claim. By the linearity of expectation, it suffices to show that for every pair of vertices u, v which exhibits a discrepancy under π before the blowup, the expected number of discrepancies of the two corresponding clusters in G' and H' is under $(n'/n)^2 < \binom{n'}{2} \Big/ \binom{n}{2}$.

Now, let $k = n' \pmod n$ and $m = \lfloor n'/n \rfloor$. The number of discrepancies due to $\{u, v\}$ is the product of the sizes of u and v's clusters (denote their sizes $\operatorname{cs}(u)$, $\operatorname{cs}(v)$). Each of these clusters has size either m or $m+1$; thus

$$\begin{aligned}
\mathbf{Ex}[\operatorname{cs}(u) \cdot \operatorname{cs}(v)] &= 1 \cdot (m \cdot m) + \mathbf{Pr}[\operatorname{cs}(u) = m+1] \cdot (1 \cdot m) \\
&\quad + \mathbf{Pr}[\operatorname{cs}(v) = m+1] \cdot (m \cdot 1) \\
&\quad + \mathbf{Pr}[\operatorname{cs}(u) = \operatorname{cs}(v) = m+1] \cdot (1 \cdot 1) \\
&= m^2 + 2 \cdot m \cdot \frac{k}{n} + \mathbf{Pr}[\operatorname{cs}(u) = \operatorname{cs}(v) = m+1] \\
&= m^2 + 2 \cdot m \cdot \frac{k}{n} + \left(\frac{k}{n} \cdot \frac{k-1}{n-1}\right) < \left(m + \frac{k}{n}\right)^2 = \left(\frac{n'}{n}\right)^2
\end{aligned}$$

This completes the proof. □

Incidentally, Pikhurko has shown in [Pik10, Lemma 14] that the distance between blowups can't be very far below the distance between the original graphs: $\mathrm{dist}\big(G', H'\big) \geq \frac{1}{3}\,\mathrm{dist}\big(G, H\big)$, for exactly-balanced blowups; this non-trivial direction of the distance bound, however, is only relevant to Chapter 4 of this work (see, specifically, Subsection 4.5.2), and not to this chapter.

Definition 3.2.5. A graph property Π is said to be (s, δ)-*inflatable* if for any graph G satisfying Π, of order at least s, all blowups of G are δ-close to satisfying Π. A property Π is said to be (s, δ)-*inflatable on the average* if for any graph G satisfying Π, of order at least s, the expected distance from Π of blowups of G to any fixed order (a uniform sampling out of all possible blowups to that order) is less than δ.

As noted above, blowups do not affect graph distances overmuch. This implies that taking a blowup cannot drive you too far away from an inflatable property:

Proposition 3.2.6. *Let property Π be (s, δ)-inflatable on the average, let G be a graph of order $n \geq s$, and let $n' > n$. The expected distance of G from the property does not increase by more than δ with a random blowup, i.e.* $\mathbf{Ex}_{G'}\big[\mathrm{dist}\big(G', \Pi\big)\big] \leq \mathrm{dist}\big(G, \Pi\big) + \delta$.

Proof. Let $H \in \Pi$ be a graph of the same order as G such that $\mathrm{dist}\big(G, \Pi\big) = \mathrm{dist}\big(G, H\big)$. Let G' and H' be corresponding random blowups of G and H respectively (as per Definition 3.2.3). The lemma gives $\mathbf{Ex}_{G'}\big[\mathrm{dist}\big(G', H'\big)\big] < \mathrm{dist}\big(G, H\big)$; also, since Π is (s, δ)-inflatable on the average, and since H if of order at least s, and since H' is a also random blowup, its own expected distance from Π is less than δ. We can now use the triangle inequality to conclude that:

$$\begin{aligned}
\mathop{\mathbf{Ex}}_{G'}\big[\mathrm{dist}\big(G', \Pi\big)\big] &\leq \mathop{\mathbf{Ex}}_{G'}\big[\mathrm{dist}\big(G', H'\big) + \mathrm{dist}\big(H', \Pi\big)\big] \\
&= \mathop{\mathbf{Ex}}_{G'}\big[\mathrm{dist}\big(G', H'\big)\big] + \mathop{\mathbf{Ex}}_{G'}\big[\mathrm{dist}\big(H', \Pi\big)\big] \\
&< \mathrm{dist}\big(G, H\big) + \delta = \mathrm{dist}\big(G, \Pi\big) + \delta
\end{aligned}$$

as claimed. □

Having defined the approximate notion of inflatability, let us make a similar definition of approximate heredity:

Definition 3.2.7. A property Π is said to be (s, δ)-*hereditary* if, for every graph in Π, all of its induced subgraphs of order at least s are δ-close to Π. A property Π is said to be (s, δ)-*hereditary on the average* if, for every graph in Π, the expected distance from Π of a uniformly-sampled subgraph of any fixed order $s' \geq s$ is less than δ.

3.2.2 Fixed-order subgraph distributions of graphs

Definition 3.2.8. Given a graph G, consider the graph induced by a uniformly sampled subset of s vertices. We denote the distribution of this induced subgraph by D^s_G, the

order-s subgraph distribution of G; $\mathrm{D}^s_G(G')$ is the relative frequency of a subgraph G' of order s in G.

Note. In [FN07b], this distribution is called the graph's *q-statistic*.

Definition 3.2.9. Let $\mathcal{G}^s$ denote the set of all graphs of order s. The distance between two distributions D, D' over graphs of order s, denoted $\mathrm{dist}\big(D, D'\big)$, is the variation distance between them, i.e.

$$\mathrm{dist}\big(D, D'\big) = \frac{1}{2} \sum_{G \in \mathcal{G}^s} \big|D(G) - D'(G)\big|$$

The distance between two graphs' distributions of order-s subgraphs cannot exceed their relative distance as graphs by more than a factor depending on s:

Lemma 3.2.10. *If two graphs G, H (of order $n \geq s$) are $\delta\binom{s}{2}^{-1}$-close, then their order-s subgraph distributions are δ-close, i.e.* $\mathrm{dist}\big(\mathrm{D}^s_G, \mathrm{D}^s_H\big) \leq \delta$.

Proof. Let $\phi : V(G) \to V(H)$ be a bijection achieving the minimum of the number of edge discrepancies. The graphs' being $\delta\binom{s}{2}^{-1}$-close means that there are at most $\delta\binom{s}{2}^{-1} \cdot \binom{n}{2}$ such discrepancies. Now consider a uniformly-sampled set of s vertices in $V(G)$, and the subgraph they induce in G and (through ϕ) in H. Every pair of vertices in the subgraph is uniformly distributed among the pairs of vertices of G or of H, so the probability of having any discrepant edges between these two subgraphs under ϕ is at most δ. When we condition on the sample not containing any vertex pair discrepant under ϕ, the distributions of such an order-s subgraph of G and of H become identical; the variation distance between the unconditioned distributions cannot, therefore, exceed δ. □

Another feature of the order-s subgraph distribution is that it does not change overmuch when taking the blowup of a graph.

Lemma 3.2.11. *Let $\delta > 0$, let G be a graph of order $n \geq \frac{2}{\delta}\binom{s}{2}$, let G' be a random blowup of G to order $n' > n$, and let $\mathcal{H} \subseteq \mathcal{G}^s$. Then*

$$\left| \mathop{\mathbf{Ex}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} [H \in \mathcal{H}] \right] - \mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_G} [H \in \mathcal{H}] \right| < \delta$$

Proof. Let $\tilde{\mathrm{D}}^s_{G'}$ denote the order-s subgraph distribution of G', conditioned on the event that every vertex of the subgraph is in the cluster of a different vertex of G. For any fixed G', we have

$$\left| \mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} [H \in \mathcal{H}] - \mathop{\mathbf{Pr}}_{H \sim \tilde{\mathrm{D}}^s_{G'}} [H \in \mathcal{H}] \right| \leq \mathrm{dist}\big(\mathrm{D}^s_{G'}, \tilde{\mathrm{D}}^s_{G'}\big)$$

This variation distance is bounded by the probability p that multiple vertices in H sampled uniformly from G' are in the same cluster of vertex of G. For a given pair of vertices of H, the probability of their being in the same cluster is at most the relative size of a large cluster, which is bounded by $2/n$; union-bounding over all pairs, we have, irrespective of G',

$$p < \binom{s}{2} \cdot \frac{2}{n} \leq \binom{s}{2} \cdot \frac{2}{\frac{2}{\delta}\binom{s}{2}} = \delta$$

The proof can now be complete if we show that

$$\mathop{\mathbf{Ex}}_{G'}\left[\mathop{\mathbf{Pr}}_{H\sim \tilde{\mathrm{D}}^s_{G'}}[H \in \mathcal{H}]\right] = \mathop{\mathbf{Pr}}_{H\sim \mathrm{D}^s_G}[H \in \mathcal{H}]$$

For this purpose, let us analyze separately the various sets of s vertices in G (corresponding to sets of s clusters in G'): The probability of sampling H in $\mathcal{H}$ is the probability of sampling a set S of s vertices, such that the induced graph $H = H_S$ on these vertices is in $\mathcal{H}$; in G', it is the probability of sampling vertices from the appropriate sets of s clusters. Let $\mathcal{S}_\mathcal{H}$ be the family of s-vertex sets S with $H_S \in \mathcal{H}$. Denote by $p_S(G')$ the probability that a set S' of s vertices, each from a different cluster of a G vertex, equals S. Now, by the linearity of expectation,

$$\mathop{\mathbf{Ex}}_{G'}\left[\mathop{\mathbf{Pr}}_{H\sim \tilde{\mathrm{D}}^s_{G'}}[H \in \mathcal{H}]\right] = \mathop{\mathbf{Ex}}_{G'}\left[\sum_{S\in\mathcal{S}_\mathcal{H}} p_S(G')\right] = \sum_{S\in\mathcal{S}_\mathcal{H}} \mathop{\mathbf{Ex}}_{G'}\big[p_S(G')\big]$$

The expectation $\mathbf{Ex}_{G'}[p_S(G')]$ is the same, by symmetry, for all s-subsets S, as the blowup G' is sampled uniformly. It must therefore be equal to the inverse of the number of sets S, i.e. $\binom{n}{s}^{-1}$. Thus

$$\mathop{\mathbf{Ex}}_{G'}\left[\mathop{\mathbf{Pr}}_{H\sim \tilde{\mathrm{D}}^s_{G'}}[H \in \mathcal{H}]\right] = \sum_{S\in\mathcal{S}_\mathcal{H}} \mathop{\mathbf{Ex}}_{G'}\big[p_S(G')\big] = \sum_{S\in\mathcal{S}_\mathcal{H}} \binom{n}{s}^{-1} = \mathop{\mathbf{Pr}}_{H\sim \mathrm{D}^s_G}[H \in \mathcal{H}]$$

as claimed. □

Note that while a single event (or a single order-s subgraph or set of s clusters) has the same expected probability when taking a random blowup, in specific blowups the probability of an event or a set of clusters may very well be quite different, even for $n \gg s$, as one may choose to have, say, the higher-degree vertices have bigger clusters, and the lower-degree vertices have smaller clusters. The following proposition gives a deterministic bound on the distance between the subgraph distributions using both the order of the pre-blowup graph n and the 'imbalance' of the blowup:

Proposition 3.2.12. *Let G be a graph of order $n \geq s$ and G' a blowup of G to order*

$n' \geq n$, *and let* $k = n' \pmod{n}$. *If* n *divides* n', *then*

$$\operatorname{dist}\big(\mathrm{D}^s_{G'}, \mathrm{D}^s_G\big) < \binom{s}{2} \cdot \frac{1}{n}$$

and for any $n \geq n'$ *it holds that*

$$\operatorname{dist}\big(\mathrm{D}^s_{G'}, \mathrm{D}^s_G\big) < \binom{s}{2} \cdot \frac{1}{n} + s \cdot \frac{\min\{k, n-k\}}{n'} < \binom{s}{2} \cdot \frac{1}{n} + s \cdot \frac{n}{n'}$$

Proof. Let us first analyze the case of the blowup G' being exactly-balanced, i.e. $n' = n \cdot k$ for some $k \in \mathbb{N}$. Consider a sample of an s-vertex subgraph of G'. Conditioning on the event of every vertex being sampled from the cluster of a different vertex of G, the distribution of order-s subgraphs of G' is exactly D^s_G. Thus the unconditioned distance $\operatorname{dist}\big(\mathrm{D}^s_{G'}, \mathrm{D}^s_G\big)$ is at most the probability of sampling at least two of the s vertices from the same cluster. Since G' is an exactly-balanced blowup, this probability is less than $1/n$ for a single pair of vertices. Applying a union bound over the $\binom{s}{2}$ pairs of vertices yields $\operatorname{dist}\big(\mathrm{D}^s_{G'}, \mathrm{D}^s_G\big) < \frac{1}{n}\binom{s}{2}$.

In the general case, G' is not necessarily exactly-balanced. However, let us choose one vertex from each of the $n' \pmod{n}$ larger clusters to form a set U. the subgraph of G' induced by $V(G') \setminus U$ is an exactly-balanced blowup of G; and with probability at least $1 - s \cdot \frac{k}{n'}$, a sample of s vertices from $V(G)$ is in fact sampled from $V(G') \setminus U$ only, conditioning on which event the above distance bound holds. Alternatively, think of an exactly-balanced blowup G'' of G, to order $n' + n - k$. The exactly-balanced distance holds for G'', but when conditioning on the event of no vertices being sampled out of the $n - k$ additional vertices in G'', it has the same order-s subgraph distribution as G'; this event's probability is at least $1 - s \cdot \frac{n-k}{n'}$.

In the general case, therefore, we have

$$\operatorname{dist}\big(\mathrm{D}^s_{G'}, \mathrm{D}^s_G\big) < \min\left\{\frac{1}{n}\binom{s}{2} + \frac{k}{n'}, \frac{1}{n}\binom{s}{2} + \frac{n-k}{n'}\right\}$$

as claimed. □

3.3 Overview of results

We first state our main result in a simplified manner, for motivation and clarity:

Theorem 3.1. *If a hereditary, inflatable graph property has a test making* $q(\varepsilon)$ *queries, regardless of the size of the input graph, then it has a strongly canonical test — specifically, a natural test — making* $\mathrm{O}\big(q(\varepsilon)^4\big)$ *queries.*

We will in fact prove a mildly stronger version, with the above being a special case:

Theorem 3.1 (exact version)**.** *Let* Π *be a graph property that has a test with queries involving at most* $s(\varepsilon)$ *distinct vertices, regardless of the size of the input graph, and let*

$s_1 = 12\binom{31s}{2}$. *If* Π *is* $\left(s_1, \frac{1}{6}\binom{s_1}{2}^{-1}\right)$*-hereditary on the average and* $\left(s_1, {s_1}^{-1}\right)$*-inflatable on the average, then it has a strongly canonical test whose queried subgraph order is* $s_1 = \mathrm{O}\left(s(\varepsilon)^2\right)$.

Note. This theorem should also hold also for properties with weaker inflatability — a higher threshold value than stated above for ε-inflatability on the average — with some modifications of our proof, and with a worse dependence of the queried subgraph order on s.

We also prove a weak converse of Theorem 3.1:

Theorem 3.2. *If a graph property* Π *has a natural (not necessarily canonical) test with queries involving* $s(\varepsilon)$ *distinct vertices, then for every* $\varepsilon' > \varepsilon$, Π *is* $\left(s_h, \varepsilon'\right)$*-hereditary on the average and* $\left(s_i, \varepsilon'\right)$*-inflatable on the average, for* $s_h = \mathrm{O}\left(s \cdot \log\left(\frac{1}{\varepsilon' - \varepsilon}\right)\right)$ *and* $s_i = \mathrm{O}\left(s^2 \cdot (\varepsilon' - \varepsilon)^{-1} \log^2\left(\frac{1}{\varepsilon' - \varepsilon}\right)\right)$ *respectively (with the coefficients* s_h *and* s_i *being independent of the specific property* Π*).*

Let us now recall the proposition from Goldreich and Trevisan discussed in the introduction:

Proposition ([GT03, proposition D.2], corrected as per [GT05])**.** *Let* Π *be a hereditary graph property, with a* natural *test making* $q(\varepsilon)$ *queries. Then* Π *has a perfectly canonical (one-sided) test with queried subgraph order* $O(q(\varepsilon))$.

Originally, this proposition was stated without requiring that the test be natural (merely that the number of queries be independent of the order of the input graph). Combining now this corrected, qualified version above with Theorem 3.1, one obtains:

Corollary 3.3. *Let* Π *be a hereditary inflatable graph property, with a test making* $q(\varepsilon)$ *queries. Then* Π *has a perfectly canonical (one-sided) test with queried subgraph order* $\mathrm{O}\left((q(\varepsilon))^2\right)$.

We use the contrapositive of this corollary to provide a more straightforward proof of [AS06, Theorem 1], even improving it slightly for the case of triangles (using the recent result in [Elk11]):

Theorem 3.4. *Any* ε*-test — natural or otherwise, with one-sided or two-sided error — for the property of being triangle-free makes* $\Omega\left((1/\varepsilon)^{c\cdot(\log(1/\varepsilon))^{1+\frac{2}{\ln(2)\cdot\log(1/\varepsilon)}}}\right)$ *queries, for some global constant* c.

(The lower bound in [AS06, Theorem 1], is $(c/\varepsilon)^{c\cdot\ln(c/\varepsilon)}$.)

Returning to [GT03, proposition D.2], while for hereditary inflatable properties we have established it with a power-of-four penalty on the number of queries, for properties with one-sided tests it can be shown to hold as stated:

Proposition 3.3.1. *If a hereditary inflatable property* Π *has a one-sided (not necessarily natural) test making* $q(\varepsilon)$ *queries, then* Π *has a perfectly canonical test with queried subgraph order at most* $2q(\varepsilon)$.

Finally, we place the notion of inflatability in the context of proximity-oblivious testing (see the exposition of this concept in Section 3.8), we prove the following partial characterization:

Proposition 3.3.2. *Let* Π *be an inflatable hereditary property.* Π *has a constant-query, proximity-oblivious test if and only if there exists a constant* s *such that, for* $n \geq s$, Π_n *consists exactly of those graphs of order* n*, which are free of order-*s *graphs outside of* Π_s.

3.4 Naturalizing tests

In this section we prove Theorem 3.1.

Let Π be a property meeting the conditions in the statement of the theorem. As Π has a test with queries involving at most $s(\varepsilon)$ vertices (independently of n), by [GT03, Theorem 2] it has a canonical test, querying a uniformly sampled subgraph of order at most $9s$, in its entirety. As noted after the citation of this theorem, in Subsection 3.2.1 above, we may assume that the canonical test's probability of error is at most $\frac{1}{36}$ rather than $\frac{1}{3}$, at the cost of increasing the queried subgraph order to $s_0 = 31s$.

One may think of the existence of such a canonical test as meaning that the membership of a graph in Π is essentially determined by its distribution of (induced) subgraphs of order s_0. This being the case, let us consider a (canonical) 'meta-test' for Π, which estimates whether the subgraph distribution leads to acceptance (of the input graph G of order n). This meta-test is listed as Algorithm 3.1.

Note. The order s_1 of the larger subgraph used for this estimate is chosen so as to ensure the stability of the distribution under blowups — a consideration which will become relevant later in this section. On the other hand, s_1 is not high enough to properly *estimate the distribution*, i.e. estimate the frequency of specific order-s_0 subgraphs (there are $\exp\big(\Omega\big({s_0}^2\big)\big)$ of them) in G.

Algorithm 3.1 A Meta-Test for Π

1: Uniformly query a subgraph G_{sample} of order $s_1 = 12\binom{s_0}{2} = 12\binom{31s(\varepsilon)}{2}$.
2: If at least a $\frac{1}{6}$-fraction of the order-s_0 subgraphs G' of G_{sample} are such that the (canonical) s_0-test accepts G by sample G', accept. Otherwise reject.

Lemma 3.4.1. *Algorithm 3.1 is a valid test for property* Π*, with probability of failure at most* $1/6$.

Proof. Suppose the input graph G either satisfies Π or is ε-far from satisfying Π. Let G' be one of the $\binom{s_1}{s_0}$ order-s_0 subgraphs of G_{sample}. Let $X_{G'}$ be the indicator for the s_0-test erring (that is, rejecting G in case G satisfies Π, or accepting G in case G is far from Π) by sample G'. Every order-s_0 subgraph of G_{sample} is in fact uniformly sampled from the input graph, thus $\mathbf{Ex}[X_{G'}]$ is the probability of the s_0-test erring — at most $\frac{1}{36}$. The expected fraction of order-s subgraphs of G_{sample} by which the s_0-test errs is therefore also at most $\frac{1}{36}$. Considering the meta-test's behavior again, it can only err if at least a $\frac{1}{6}$-fraction of the subgraphs of G_{sample} cause the s_0-test to err. by Markov's inequality the probability of this occurring is at most $\frac{1}{36}/\frac{1}{6} = \frac{1}{6}$. □

Let us now modify Algorithm 3.1 to reject samples which are themselves not in the property at order s_1; the result is listed as Algorithm 3.2.

Algorithm 3.2 Modified Meta-Test for Π

1: Uniformly query a subgraph G_{sample} of order $s_1 = 12\binom{s_0}{2} = 12\binom{31s(\varepsilon)}{2}$.
2: If G_{sample} is not in Π, reject.
3: If at least a $\frac{1}{6}$-fraction of the order-s_0 subgraphs G' of G_{sample} are such that the s_0-test accepts G by sample G', then accept. Otherwise reject.

Lemma 3.4.2. *Algorithm 3.2 is a valid test for property* Π.

Proof. The additional check only increases the probability of rejection of any input graph, so it does not adversely affect the soundness of the modified test (that is, a graph ε-far from Π is still rejected by Algorithm 3.2 with probability at least $\frac{5}{6} \geq \frac{2}{3}$).

As for the modified test's completeness, we recall that Π is $\left(s_1, \frac{1}{6}\binom{s_1}{2}^{-1}\right)$-hereditary on the average. This implies that, for an input graph in Π, the average distance of subgraphs of order s_1 from Π is $\frac{1}{6}\binom{s_1}{2}^{-1}$; as each order-$s_1$ subgraph not in Π is at least $\binom{s_1}{2}^{-1}$-far from Π, the fraction of order-s_1 subgraphs of G which aren't in Π is at most $\frac{1}{6}$. Regardless of these, at most a $\frac{1}{6}$-fraction of the order-s_1 subgraphs of a satisfying graph cause Algorithm 3.1 to reject. Union bounding over these two sets of subgraphs causing rejection we find that the probability of the modified meta-test rejecting a graph in Π is less than $2 \cdot \frac{1}{6} = \frac{1}{3}$. □

Now, Algorithm 3.2 is not necessarily natural, receiving as input the order n of the graph G being tested, and passing this value to the original s_0-test; but if Algorithm 3.2 were somehow also natural, this would complete the proof of Theorem 3.1, as the test otherwise meets the requirements. Since Algorithm 3.2 is canonical, its naturality means being strongly canonical: accepting the same set of sampled subgraphs for any input graph order. Interestingly enough, our modification has indeed made this the case:

Lemma 3.4.3. *Let H be a graph of order s_1 by which sample Algorithm 3.2 accepts for at least some input graph order n. Algorithm 3.2 cannot reject for any input graph order $n' \geq s_1$ by sample H.*

Proof. Assume on the contrary that Algorithm 3.2 rejects by sample H for some $n' \geq s_1$. We first note that Algorithm 3.2 does not reject by H at order n' on account of H not being in Π (as samples which aren't in Π are rejected at all input orders). We will show that this invariably implies that the original test is incomplete.

Let $\Pi'_{n'}$ denote the set of order-s_0 subgraphs by which sample the s_0-test accepts an input graph G at order n'. Our assumption is that the probability of the s_0-test accepting a subgraph of H is less than $\frac{1}{6}$, or in terms of the subgraph distribution, $\mathbf{Pr}_{H_{\mathrm{s}} \sim \mathrm{D}_H^{s_0}}\left[\Pi'_{n'}\right] < \frac{1}{6}$.

Now, consider a random blowup H' of H to order n'. Π is $\left(s_1, \frac{1}{12}\binom{s_0}{2}^{-1}\right)$-inflatable on the average, and H is in Π, so

$$\mathop{\mathbf{Ex}}_{G'}\left[\mathrm{dist}\big(H', \Pi\big)\right] < \frac{1}{12}\binom{s_0}{2}^{-1}$$

and by Markov's inequality,

$$\mathop{\mathbf{Pr}}_{H'}\left[\mathrm{dist}\big(H', \Pi\big) \geq \frac{1}{6}\binom{s_0}{2}^{-1}\right] < \frac{1}{2}$$

Also, let $\delta = \frac{1}{6}$. Since $s_1 \geq \frac{2}{\delta}\binom{s_0}{2}$, we may apply Lemma 3.2.11 (substituting H and H' for G and G', s_0 for s, s_1 for n) for the event of the s_0-test accepting at order n':

$$\begin{aligned}
\mathop{\mathbf{Ex}}_{H'}\left[\mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_{H'}^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right]\right] \leq & \mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_H^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] \\
& + \left|\mathop{\mathbf{Ex}}_{H'}\left[\mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_{H'}^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right]\right] - \mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_H^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right]\right| \\
& < \mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_H^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] + \delta < \frac{1}{6} + \frac{1}{6} = \frac{1}{3}
\end{aligned}$$

and again by Markov's inequality

$$\mathop{\mathbf{Pr}}_{H'}\left[\mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_{H'}^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] \geq \frac{2}{3}\right] < \frac{1}{2}$$

Combining these two facts, we conclude that with positive probability, H' is a graph which is both very close to Π and is accepted by the s_0-test with probability at most $\frac{2}{3}$.

Now, let $\widetilde{H'}$ be a graph in Π at distance at most $\frac{1}{6}\binom{s_0}{2}^{-1}$ from H'. By Lemma 3.2.10, these two graphs' order-s_0 subgraph distributions are $\frac{1}{6}$-close, implying that

$$\left|\mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_{\widetilde{H'}}^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] - \mathop{\mathbf{Pr}}_{H_{\mathrm{s}} \sim \mathrm{D}_{H'}^{s_0}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right]\right| < \frac{1}{6}$$

We now use the triangle inequality to bound the probability of the s_0-test accepting $\widetilde{H'}$:

$$\begin{aligned}\mathop{\mathbf{Pr}}_{H_{\mathrm{s}}\sim \mathrm{D}^{s_0}_{\widetilde{H'}}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] &\leq \mathop{\mathbf{Pr}}_{H_{\mathrm{s}}\sim \mathrm{D}^{s_0}_{H'}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] + \left|\mathop{\mathbf{Pr}}_{H_{\mathrm{s}}\sim \mathrm{D}^{s_0}_{H'}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right] - \mathop{\mathbf{Pr}}_{H_{\mathrm{s}}\sim \mathrm{D}^{s_0}_{\widetilde{H'}}}\left[H_{\mathrm{s}} \in \Pi'_{n'}\right]\right| \\ &< \frac{2}{3} + \frac{1}{6} = \frac{5}{6}\end{aligned}$$

This contradicts the original test's probability of error — it must accept $\widetilde{H'}$, a graph in Π, with probability at least $1 - \frac{1}{36} > \frac{5}{6}$. It can therefore not be the case that Algorithm 3.2 rejects H at order n'. □

Proof of Theorem 3.1. Given a property Π satisfying the conditions, we have devised Algorithm 3.2: This is a canonical test for Π, with queried subgraph order $s_1 = 12\binom{31s}{2}$; by Lemma 3.4.3, it accepts and rejects the same set of queried subgraphs for all graph orders $n \geq s_1$ — that is, it is a natural test. □

3.5 Lower bounds for triangle-freeness testing

As discussed earlier, part of our interest in the naturalization of tests is obtaining lower bounds on testing the property of triangle-freeness (or freeness of other induced substructures), through lower bounds on one-sided testing or other more fundamental results.

The current state of the art in terms of an explicitly-stated lower bound is:

Theorem ([AS06, Theorem 1])**.** *The query complexity of any ε-test — natural or otherwise, with one-sided or two-sided error — for the property of being triangle-free is at least $(c/\varepsilon)^{c\cdot\ln(c/\varepsilon)}$, for some global constant c.*

Now, consider the contrapositive of Corollary 3.3:

Corollary. *If a hereditary inflatable property has no perfectly canonical test with queried subgraph order $q'(\varepsilon)$, then it has no test whatsoever (natural or otherwise, with one-sided or two-sided error) making $q(\varepsilon)$ queries such that $q(\varepsilon)^2 = o(q'(\varepsilon))$.*

[AS06, Theorem 1] can be obtained by combining the one-sided lower bound for testing triangles of [Alo02] with Corollary 3.3, without requiring the careful use of Yao's method in [AS06].

The proof of the one-sided testing lower bound, in [Alo02], is based on a construction of a large subset of $[n]$, which is free of arithmetic progressions (i.e. tuples $x, x + d, x + 2d, x + 3d, \ldots$). The specific construction used in [Alo02] is that of Behrend, in [Beh46]. Recently, after 60 years with no progress, an improvement was made over this construction by Michael Elkin in [Elk11] (with a simpler proof suggested by Green and Wolf in [GW10]):

Theorem. *For every natural number n, there exists a subset $X_n \subseteq [n]$, with $|X_n| = \Omega\left(\frac{n \cdot \sqrt[4]{\log(n)}}{2^{\sqrt{\log(n)}}}\right)$, which contains no 3-term arithmetic progressions.*

Now, this new construction can be translated into a lower bound on testing triangle-freeness either using our methods, or using the Alon-Shapira Yao-style argument from [AS06], so that an improved two-sided lower bound can be considered to already be established as the state of the art. However, as it has not been explicitly stated in the literature, we sketch the proof below.

Lemma 3.5.1 (implicit in [Alo02] and [AS04b])**.** *Let $m(\varepsilon)$ be the highest integer with a subset $X_m \subseteq [m]$ of size εm which contains no non-trivial solutions to the equation $x_1 + , \ldots, x_{k-1} = (k-1) \cdot x_k$ (for an odd k). Any one-sided-error test for the property of a graph being k-cycle-free makes $\Omega\big((m(\varepsilon))^{k-2}\big)$ queries.*

Proof Sketch. One constructs a k-partite graph of size $\Theta(m(\varepsilon))$, and connects vertex i in each of the first $k-1$ parts to each vertex in the set $\{i + x \mid x \in X\}$ in the next part, for every i. One then connects the vertex i of the k^{th} part with each vertex in the set $\{i - kx \mid x \in X\}$, for every i. It can be shown that this graph has $\Theta(m|X|)$ k-cycles, all distinct — as two k-cycles can only share an edge if X has a k-term arithmetic progression. As $|X| > \varepsilon m$, the graph is far from being k-cycle-free.

One then blows up the graph by a factor of $\Theta(n/m)$. The resulting graph can be shown to be far from being k-cycle-free, but only has $\Theta\big((n/m)^k \cdot m \cdot |X|\big) = O\big(n^k/m^{k-2}\big)$ cycles. Now, a one-sided test making $o\big(m^{k-2}\big)$ queries will not find any of these cycles in the blown-up graph, and will have to accept (as its queries can be completed into a k-cycle-free graph).

(This argument, with some modification and for the case of 4-cycles in digraphs, is made in detail in Chapter 5.)

Observation 3.5.2 ([Alo02])**.** If a set of integers is free of 3-term arithmetic progressions, then it is free of solutions to the equation $x_1 + x_2 = (3-1)x_3$.

Combining Lemma 3.5.1 and Observation 3.5.2 with the contrapositive form of Corollary 3.3, we have, for the case of graphs:

Corollary 3.5. *Let $m(\varepsilon)$ be the highest integer with a subset $X_m \subseteq [m]$ of size εm which is free of 3-term arithmetic progressions. Any test for the property of a graph being triangle-free makes $\Omega\big((m(\varepsilon))^{1/4}\big)$ queries.*

Now, the progression-free set used in [Alo02] has size

$$|X_n| > \frac{n}{\exp\Big(10\sqrt{\ln(n)\ln(k)}\Big)}$$

which implies $m(\varepsilon) \geq (c/\varepsilon)^{c \cdot \ln(c/\varepsilon)}$, for an appropriate global constant c'. The Elkin construction has size

$$|X_n| = \Omega\left(n \cdot \frac{\sqrt[1/4]{\log(n)}}{2^{\sqrt{\log(n)}}}\right)$$

with $\log(n)$ being the base-2 logarithm, implying that $m(\varepsilon) \geq \exp\big(c' \cdot \log^{2p}(1/\varepsilon)\big)$, for an appropriate global constant c, and with $p = 1 + \frac{1}{\ln(2) \cdot \log(1/\varepsilon)}$ (we omit the calculation.) This proves Theorem 3.4.

Note. A generalization of Elkin's result by Kevin Obryant to k-progression-free sets in [Obr11] hints at possible similar lower bounds on testing induced k-cycle freeness. However, the argument in Observation 3.5.2 does not apply to cycles of length over 3 (e.g. $1 + 3 + 5 + 7 = (5 - 1) \cdot 4$ is a 5-term linear equation, but the set $\{1, 3, 4, 5, 7\}$ has no 5-term progression); one would have avoid cycles due to such solutions in an alternative construction.

3.6 One-sided error and natural tests

Observation 3.6.1. If a hereditary property has a strongly canonical test, then this test must be one-sided.

Proof. If the test for the hereditary property Π (deterministically) rejects any sampled subgraph G' of a graph $G \in \Pi$, the test also rejects G' when it is the entire graph. But when G' is the entire graph, it will always be the sampled subgraph, i.e. the test rejects G' with probability 1. G' can therefore not be in Π — a contradiction to Π being hereditary. □

The implication in Observation 3.6.1 can be reversed, in a way — weak approximate heredity as a consequence of one-sided testability:

Lemma 3.6.2. *If a property Π has a one-sided strongly canonical test with queried subgraph order $s(\varepsilon)$ for some ε, then Π is $\big(s(\varepsilon), \varepsilon\big)$-hereditary.*

Proof. Let $G \in \Pi_n$ for $n \geq s(\varepsilon)$, and let G' be a subgraph of G of order at least $s(\varepsilon)$. If G' is ε-far from Π, then it must have an order-s subgraph G'' by which sample the test rejects G'. But the test also rejects G by sample G'', in contradiction to its one-sidedness. □

Note. This lemma is somewhat similar to the second direction of [AS08a, Theorem 2], in which the existence of a one-sided natural test is shown to imply 'semi-heredity'.

One would hope to somehow get rid of the dependence on ε and find conditions under which the property is hereditary, at least down to some n_0; this becomes possible if

the test is proximity-oblivious, but note that if a property Π has a natural proximity-oblivious test, then Π is simply the property of being free of subgraphs by which this test rejects (at least for $n \geq s$; see discussion in Section 3.8, and specifically Proposition 3.3.2).

In the proof of Lemma 3.6.2, we used the one-sidedness of the test to obtain deterministic approximate heredity; Section 3.7 below deals with the general, two-sided case, and establishes approximate heredity only on the average. Deterministic approximate heredity may indeed require the test to be one-sided. For example, the property Π_{half}, containing those graphs with at most $\frac{1}{2}\binom{n}{2}$ edges, is $\left(O\left(\frac{1}{\delta}\right), \delta\right)$-hereditary on the average, has a two-sided natural test (in fact, its query complexity can be shown to be $O\left(1/\varepsilon^2\right)$), but it is not $\left(s, \frac{1}{2} - \delta\right)$-hereditary for any s and $\delta > 0$ (as there are satisfying graphs with arbitrarily large complete subgraphs).

Returning again to the direction of Theorem 3.1, let us follow an alternate line of argumentation than the one used to prove Theorem 3.1, this time for the case of one-sided tests.

Lemma 3.6.3. *Let Π be an inflatable property. A one-sided canonical test for Π can only reject an input graph when it samples a subgraph which is not itself in Π.*

Proof. Suppose that, for some input graph G of order n, the test samples a subgraph $G' \in \Pi$. Since Π is inflatable, there exists a blowup G'' of G' to order n such that $G'' \in \Pi$. Now, G' is an induced subgraph of G'', so it is possible for the test to sample G' when G'' is the input graph. Since the test is one-sided, it can not, therefore, reject an input graph of order n with G' as the sample. □

Proof of Proposition 3.3.1. By [GT03, Theorem 2], Π has a canonical one-sided test with queried subgraph order $s(\varepsilon) \leq 2q(\varepsilon)$, which is also one-sided. By Lemma 3.6.3, this test only rejects sampled subgraphs which are not themselves in Π. Now suppose we modify the test so as to reject all sampled subgraphs not in Π. As we are only rejecting additional subgraphs, the test's soundness can only improve. As for its completeness, we note that since Π is hereditary, no graph in Π has any subgraphs outside of Π, so the test still accepts graphs in Π with probability 1. The resulting test is indeed perfectly canonical. □

3.7 Inflatability and heredity of naturally-testable properties

Lemma 3.7.1. *If a property Π has a strongly canonical test with queried subgraph order $s(\varepsilon)$, with probability of error $\delta \leq \frac{1}{3}$, then Π is $\left(\frac{2}{\delta}\binom{s}{2}, \varepsilon + 3\delta\right)$-inflatable on the average.*

Proof. Let G be a graph of order n satisfying Π, for $n \geq \frac{2}{\delta}\binom{s}{2}$, and let G' be a random blowup of G to some higher order. Let Π' be as in Definition 3.2.1 — the set of order-s subgraphs by which sample the test accepts an input graph. By Lemma 3.2.11, we have

$$\left| \mathop{\mathbf{Ex}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} \left[H \notin \Pi' \right] \right] - \mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G}} \left[H \notin \Pi' \right] \right| < \delta$$

so $\mathbf{Ex}_{G'}\left[\mathbf{Pr}_{H \sim \mathrm{D}^s_{G'}}[H \notin \Pi']\right] < 2\delta$. By Markov's inequality

$$\mathop{\mathbf{Pr}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} \left[H \notin \Pi' \right] > 1 - \delta \right] \leq \frac{2\delta}{1-\delta} \leq 3\delta$$

Now, if G' is rejected by the test with probability at most $1-\delta$, it cannot be ε-far from Π; if it is rejected with higher probability, we can't make any assumptions regarding its distance. Thus

$$\begin{aligned} \mathbf{Ex}\left[\mathrm{dist}\left(G', \Pi\right)\right] < & \mathop{\mathbf{Pr}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} \left[H \notin \Pi' \right] \leq 1 - \delta \right] \cdot \varepsilon \\ & + \mathop{\mathbf{Pr}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} \left[H \notin \Pi' \right] > 1 - \delta \right] \cdot 1 \leq \varepsilon + 3\delta \end{aligned}$$

meeting the requirement for approximate inflatability. □

Lemma 3.7.2. *If a property Π has a strongly canonical test, with queried subgraph order $s(\varepsilon)$, with probability of error $\delta \leq \frac{1}{3}$, then Π is $\left(s, \varepsilon + \frac{3}{2}\delta\right)$-hereditary on the average.*

Proof. Let G be a graph in Π of order at least s, let G' a uniformly-sampled subgraph of G of order $s' \geq s$, and let $p_{G'}$ denote the probability of the test rejecting with G' rather than G as its input graph. The expectation of $p_{G'}$ is exactly δ, the probability of the test rejecting G — as the process of sampling an order-s' subgraph, then sampling an order-s subgraph out of it, is the same as just sampling an order-s subgraph of G. We can apply Markov's inequality and bound the probability of $p_{G'}$ being too high: $\mathbf{Pr}_{G'}[p_{G'} \geq 1-\delta] \leq \frac{\delta}{1-\delta}$. Since the test is sound, we know that if $p_{G'}$ is lower than $1-\delta$, then G' cannot be ε-far from Π; if $p_{G'}$ is higher, we do not assume anything about G''s distance from Π. Thus

$$\begin{aligned} \mathop{\mathbf{Ex}}_{G'}\left[\mathrm{dist}\left(G', \Pi\right)\right] &\leq \mathop{\mathbf{Pr}}_{G'}[p_{G'} < 1-\delta] \cdot \varepsilon + \mathop{\mathbf{Pr}}_{G'}[p_{G'} \geq 1-\delta] \cdot 1 \\ &\leq 1 \cdot \varepsilon + \frac{\delta}{1-\delta} \cdot 1 = \varepsilon + \frac{\delta}{1-\delta} \leq \varepsilon + \frac{3}{2}\delta \end{aligned}$$ □

Proof of Theorem 3.2. Let $\delta = \frac{1}{3}(\varepsilon' - \varepsilon)$. Our first step is the same as in the proof of Theorem 3.1 — pre-amplifying the probability of success of the natural test and

canonicalizing it. Our modified test remains natural (thus being strongly canonical), with probability of failure at most δ, and its queried subgraph size is $s_{\mathrm{h}} = O\big(s \cdot \log\big(\delta^{-1}\big)\big)$, as per the discussion of canonicalization in Subsection 3.2.1. Now, by Lemma 3.7.2, Π is $\big(s_{\mathrm{h}}, \varepsilon + \frac{3}{2}\delta\big)$-hereditary on the average, and by Lemma 3.7.1, Π is $\big(\frac{2}{\delta}\binom{s_{\mathrm{h}}}{2}, \varepsilon + 3\delta\big)$-inflatable on the average. This meets the claim. □

3.8 Natural testability and proximity-oblivious testing

In most works regarding property testing, tests are devised based on a foreknowledge of the proximity parameter ε: Either the test is given ε as input, or ε is fixed globally. Goldreich and Ron explore an alternative approach in [GR09]:

Definition 3.8.1. A *proximity-oblivious* test for property Π with *detection probability* $\rho(\cdot)$ is a probabilistic oracle machine, which is given the value n, as well oracle access to a graph G of order n in the same manner as a usual test. The machine accepts a graph $G \in \Pi_n$ with probability 1, and rejects a graph $G \notin \Pi_n$ with probability at least $\rho\big(\mathrm{dist}\big(G, \Pi_n\big)\big)$.

Notes.
- One can obtain an ε-test in the usual sense by invoking the proximity-oblivious test $\Theta(1/\rho(\varepsilon))$ times.
- A proximity oblivious test has query complexity which may depend on n, but not on ε.

In this section we concern ourselves with proximity-oblivious tests, that havequery complexity independent of n.

Lemma 3.8.2. *If a hereditary, inflatable graph property has a proximity-oblivious test making c queries, using $s \leq 2c$ sampled vertices, then it has a perfectly canonical proximity-oblivious test with queried subgraph order s (making at most $\binom{s}{2}$ queries).*

The proof of this lemma is exactly the proof of Proposition 3.3.1, which does not make any assumptions regarding the test's use of the value of ε, nor regarding its probability of rejecting far graphs.

The general results of [GR09] regarding the dense graph model include a characterization of the properties admitting a (not necessarily natural) constant-query proximity-oblivious test:

Theorem ([GR09, Theorem 4.7])**.** *A property Π has a constant-query proximity-oblivious test if and only if there exists a constant c and a finite sequence $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ of sets of graphs, such that each $\mathcal{F}_n$ contains graphs of size at most c, and Π_n is the set of order-n $\mathcal{F}_n$-free graphs.*

When limiting our focus to properties which we know to be naturally testable, we can tighten the characterization:

Proof of Proposition 3.3.2. If Π is the property of being $\mathcal{F}$-free for $\mathcal{F} = (\Pi_s)^c$, then Π is proximity-oblivious testable with a constant number of queries: As established by Alon, Fischer, Krivelevich and Szegedy in [AFKS00], any graph G is either close to being $\mathcal{F}$-free, or has $\delta(\varepsilon) \cdot n^s$ induced copies of this forbidden subgraph (with δ being a double-tower function of $(1/\varepsilon)$, as this fact is established using a version of Szemerédi 's regularity lemma). In this direction, our argument is the same as in the proof of the general characterization theorem of proximity-oblivious-testable properties [GR09, Theorem 4.7].

The other direction follows from Lemma 3.8.2: The existence of a proximity-oblivious test implies the existence of a perfectly canonical test querying a subgraph of order s and rejecting if it isn't in Π_s. This test accepts, with probability 1, exactly those graphs which are free of induced subgraphs outside Π_s; as it is one-sided, this implies that Π, at order s and above, is the set of $(\Pi_s)^c$-free graphs. □

3.9 Naturalization and inflatability in other dense structures

The results of this chapter all essentially hold, albeit with different parameters, for any class of dense structures which fits the general definition in Subsection 2.1.1 — and also for structures mentioned there which require some trivial reduction to fit that definition, such as matrices and tensors with no order on their coordinates in each dimension.

There is, however, a subtle point regarding the orders of structures tested: In graphs, a test whose queries involve $s(\varepsilon)$ vertices, when applied to a graph of order under s, can simply query the entire graph and decide deterministically — using a number of queries bounded by $\binom{s}{2}$. This is not generally possible in multi-partite dense structures: A test might require more vertices than are present in one of the parts, but it cannot query the entire graph without making a number of queries depending on other n_i's, which is not bounded. Instead, the test may require complex behavior, different than for the general case, to effectively test structures with some parts being small and others large. While such behavior is worthy of independent study, we wish to make straightforward generalizations of this chapter's results, so we choose to ignore this setting. We will therefore only be generalizing our results to uniform-order tests; and this choice also motivates the limited scope of our definition of canonicality in Definition 2.2.3.

We shall not repeat the proofs made above for graphs also for the case of general dense structure classes, but rather state the generalized results and provide proof sketches.

3.9.1 Generalized preliminaries

For the rest of this section, we fix a class of general dense structures (as per the definition in Subsection 2.1.1), letting k denote the number of vertex parts, t the number of edge relations, and r_i the arity of the i^{th} edge relation. We also denote $r = \max\{r_1, \ldots, r_t\}$.

Observation 3.9.1. Under our assumptions and by Definition 2.1.10, a dense structure of uniform order s supports up to $\sigma(s,t,k,r) = t \cdot (ks)^r$ potential hyperedges; if the class of structures is unconstrained, and $r_1 = \ldots = r_t = r$, then the structure supports exactly this number.

Theorem 3.6 (Generalization of [GT03, Theorem 2])**.** *If a property Π has a uniform-order test making $q(\varepsilon)$ queries involving at most $s(\varepsilon)$ vertices from each part of the input structure, independently of the size of the input structure and its parts, then Π has a canonical test, sampling a substructure of order at most $9s(\varepsilon)$ (and making $O(\sigma(9s,t,k,r)) = O(s^r) = O(q^r)$ queries). If the original test is one-sided, then a queried subgraph of order $s(\varepsilon)$ will suffice for such a canonical test, which will also be one-sided.*

Proof Sketch. The transformation of an arbitrary graph test into a canonical one in [GT03, Section 4] has three steps:

- First, the test is split into two phases: A uniform sampling of vertices, followed by a (probabilistic) decision based on their induced subgraph, queried in its entirety;
- The second phase of the test is made independent of the labeling of the vertices of the induced subgraph. In other words, the test is made to accept with the same probability any two induced subgraphs seen in the second phase which are isomorphic to each other.
- Finally, the probabilistic aspect of the second phase is discarded by rounding probabilities, so that induced subgraphs are deterministically either accepted or rejected.

Considering these three steps, one observes that they do not depend on a graph's having two vertices per edge, or on the non-partiteness of general graphs. We can therefore apply the same transformation to a test of any dense structure: We sample $O(s(\varepsilon))$ vertices from every part, and query the entire induced substructure on the sampled vertices (making $\sigma(s,t,k,r)$ queries). A deterministic decision is now be made based on this order-s substructure.

The only point one must take into account when canonicalizing tests of uniform-order partite structures is, that the choice of part from which to sample the next vertex may depend on previous query results — an aspect missing in the case of graphs. This is the reason why as many as $k \cdot s$ vertices (the number of vertices in a substructure of uniform order s) may be required: Instead of adaptively sampling s vertices, choosing

one part or another for each of them, we sample s vertices from every part, and can thus simulate the original test's sampling using our already-sampled vertices.

One may verify that the rest of the details of the proof of [GT03, Theorem 2] indeed hold regardless of the choice of general dense structure class (but assuming that the structure has enough vertices). The constant factor 9 is due to the repetition of the original test to amplify the probability of success, an amplification necessary for rounding the acceptance probabilities (and unnecessary for the case of one-sided tests). This too is the same for any dense structure.

Definition 3.9.2. For a dense structure G in our chosen class, We denote by D^s_G the distribution of substructures induced by a uniformly-sampled set of s vertices in each part — the *order-s substructure distribution* of G; $\mathrm{D}^s_G(G')$ is the relative frequency of a substructure G' of order s in G.

We let $\mathcal{G}^s$ denote all structures of uniform order s in our class of dense structures, and define the distance between distributions similarly to the case of subgraph distributions (see Definition 3.2.9).

Lemma 3.9.3 (Generelization of Lemma 3.2.10)**.** *If two dense structures G, H are $\delta/\sigma(s,t,k,r)$-close, then their order-s substructure distributions are δ-close, that is,* $\mathrm{dist}\big(\mathrm{D}^s_G, \mathrm{D}^s_H\big) \leq \delta$.

Proof Sketch. The proof is the same as in the case of graphs, except that the number of potential hyperedges in an order-s substructure is bounded by $\sigma(s,t,k,r)$ rather than $\binom{s}{2}$.

Lemma 3.9.4 (Generelization of Lemma 3.2.11)**.** *Let $\delta > 0$, let G be a structure with $n_i \geq \frac{2}{\delta} k\binom{s}{2}$, for all $i \in [k]$; let G' be a random blowup of G to some higher order $(s_1, s_2, \ldots, s_k)$ (or uniform order s); and let $\mathcal{H} \subseteq \mathcal{G}^s$. Then*

$$\left| \mathop{\mathbf{Ex}}_{G'} \left[\mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_{G'}} [H \in \mathcal{H}] \right] - \mathop{\mathbf{Pr}}_{H \sim \mathrm{D}^s_G} [H \in \mathcal{H}] \right| < \delta$$

Proof Sketch. The difference in this proof from the case of graphs is that there are as many as s vertices in each part of each structure in $\mathcal{G}^s$, so one must union-bound over as many as $k\binom{s}{2}$ pairs of vertices which may be sampled from the same cluster, rather than $\binom{s}{2}$ in graphs or other non-partite structures. Otherwise the proof is the same.

3.9.2 Generalization of our main results

Theorem 3.7 (Generalization of Theorem 3.1)**.** *If a hereditary inflatable property has a uniform-order test making $q(\varepsilon)$ queries, regardless of the size of the input structure and its parts, then it has a strongly canonical uniform-order test — specifically, a natural test — making $\mathrm{O}\big(q(\varepsilon)^{2r}\big)$ queries.*

Theorem 3.7 (Exact version)**.** *Let* $s : \mathbb{R}^+ \to \mathbb{N}$*. There exist* $s_1 = O\big(k\binom{s}{2}\big)$*,* $\varepsilon_i = \Omega(1/\sigma(s,t,k,r))$ *and* $\varepsilon_h = \Omega(1/\sigma(s_1,t,k,r))$ *for which the following holds: Suppose a property* Π *of dense structures of a certain kind is* $\big(s_1,\varepsilon_h\big)$*-hereditary on the average and* $\big(s_1,\varepsilon_i\big)$*-inflatable on the average, and that* Π *has a uniform-order test making queries involving at most* $s(\varepsilon)$ *distinct vertices in each part of the input structure (regardless of the size of the parts). Then* Π *has a strongly canonical uniform-order test querying a substructure of order* s_1*.*

Proof Sketch. The proof for the case of graphs works for whatever dense structure we are concerned with: We canonicalize the original test; switch to estimating the acceptance probability of the canonical test over a larger (order-s_1) substructure; and finally reject if the larger substructure is itself not in Π. Using Lemma 3.9.3 and Lemma 3.9.4, analysis shows that this is both a valid test and that it is natural, i.e. the same property set of sampled substructures is accepted at any input order.

The only adjustments are in the larger sampled substructure size and the heredity and inflatability parameters:

- The sampled substructure must be high enough for Lemma 3.9.4 to yield a sufficiently small constant difference in the distributions of order-s substructures; for our dense structures this is $O\big(k\binom{s}{2}\big)$ instead of the $O\big(\binom{s}{2}\big)$ for the case of graphs, as discussed in the proof of Lemma 3.9.4.

- The heredity parameter must relate to the larger substructure size s_1 as per the above. Also, it must be strong enough so that, on the average, an order-s_1 substructure of a structure in Π will itself be in Π, rather than just being close to Π; this explains the inverse dependence on the number of edges/hyperedges in the substructure.

- The inflatability parameter must be such that a random blowup of a graph in Π is close enough to Π for Lemma 3.9.3 to yield a small constant distance between the order-s substructure distributions.

The parameters appearing in the statement of the generalized theorem (for uniform-order structures) indeed meet these requirements.

The converse of Theorem 3.7 also admits exactly the same proof as for the case of graphs, with a tweaking of the inflatability parameter s_i similarly to Lemma 3.9.4:

Theorem 3.8 (Generalization of Theorem 3.2)**.** *If a property* Π *has a natural (not necessarily canonical) test which, for structures of order at least* $s(\varepsilon)$*, makes queries involving at most* $s(\varepsilon)$ *distinct vertices in each part, then for every* $\varepsilon' > \varepsilon$*,* Π *is* $\big(s_h,\varepsilon'\big)$*-hereditary on the average and* $\big(s_i,\varepsilon'\big)$*-inflatable on the average, for* $s_h = \mathrm{O}\big(s \cdot \log\big(\frac{1}{\varepsilon'-\varepsilon}\big)\big)$ *and* $s_i = O\Big(ks^2 \cdot (\varepsilon'-\varepsilon)^{-1}\log^2\big(\frac{1}{\varepsilon'-\varepsilon}\big)\Big)$ *respectively.*

Note. In this direction, we are not limiting the argument to uniform-order tests.

Alon's [GT03, Proposition D.2] regarding perfectly canonical testing of hereditary properties (quoted above with its qualification in [GT05]) applies, with the same proof, to any class of dense structures; with it, and Theorem 3.7, we derive the following:

Corollary 3.9 (Generalization of Corollary 3.3)**.** *If a property* Π*, which is hereditary and inflatable, has a uniform-order test making* $q(\varepsilon)$ *queries, then it has a canonical uniform-order test with queried subgraph order* $\mathrm{poly}(q(\varepsilon))$*.*

Proposition 3.9.5 (Generalization of Proposition 3.3.1)**.** *If a property* Π*, which is hereditary and inflatable, has a one-sided (not necessarily natural) uniform-order test making* $q(\varepsilon)$ *queries, then* Π *has a perfectly canonical uniform-order test with queried subgraph order at most* $r \cdot q(\varepsilon)$*.*

Chapter 4

Query complexity hierarchies for dense graphs and other models

4.1 Introduction

While the rest of this thesis is mainly concerned with properties whose query complexity depends only on the distance parameter ε, this chapter focuses on properties whose tests require more queries as graphs grow, and with this dependence of their query complexity on n.

Goldreich, Goldwasser and Ron's initial exposition of graph property testing already considered the question of properties of dense structures that are 'maximally'-dependent on n: [GGR98, Proposition 4.1.1] establishes the existence of a property of strings, or generic functions (from $[n]$ to a finite domain), with $\Omega(n)$ query complexity, linear in the size of the representation; and in [GGR98, Proposition 10.2.3.1], this construction is built upon to establish the existence of a dense graph property with query complexity linear in the size of the representation, i.e. $q(n) = \Omega\big(n^2\big)$.

There is no reason to assume a gap in the query complexity anywhere on the 'spectrum' between $q(n) = \Theta(1)$ and $q(n) = \Theta\big(n^2\big)$, especially as over time, properties have been established to have all manners of specific query complexities in between: Graph isomorphism testing, in different variants, has been shown by Fischer and Matsliah in [FM06] to have query complexities such as $\tilde{\Theta}\big(n^{3/2}\big)$ and $\tilde{\Theta}(\sqrt{n})$; Dyck languages (parenthesis languages) have been shown to require $\Omega\big(n^{1/11}\big)$ queries and be testable with $\Omega\big(n^{2/3}\mathrm{polylog}(n)\big)$ queries; et cetera. Indeed, it is natural to expect there exist properties of dense graphs (or other dense structures) with any arbitrary query complexity as a function of n: Properties testable with $\Theta(q(n))$ queries, without being testable with $o(q(n))$.

In this chapter we prove the existence of such query complexity hierarchies for three testing models: Beginning with the simple case of properties of generic Boolean functions (or equivalently, of binary strings); making an aside for the case of bounded-degree (sparse) graphs; and finally focusing on dense structures, specifically dense graphs. For

each model, we provide explicit (probabilistic) constructions for such properties. In fact, all of these hierarchy results are established in a very similar pattern:

- We start with an appropriate maximally-hard property Π' for our specific setting.
- A property Π^q is constructed for an arbitrary choice of $q(n)$, using mostly some sort of replication or blowup, so that every structure in ${\Pi^q}_n$ corresponds to some structure in Π' of size $q(n)$ or lower.
- Testing Π' is shown to be reducible (either generally, in the sense of Definition 2.4.1, or for some subset or distribution) to testing Π^q, establishing an $\Omega(q(n))$ lower bound on the query complexity of Π^q.
- A test for Π^q, making $O(q(n))$ queries, is explicitly presented, which essentially considers (for an input structure of order n), which smaller structure(s) from Π' is the input a blowup or a replication of, if at all.

There is, however, some subtlety to the question of the existence of properties of arbitrary query complexity, and even the existence of maximally-hard properties.

A first aspect to consider in this respect is the kinds of properties we wish to obtain. A "purely random" property will almost surely be hard to test, but it will also be hard to decide (and impossible to decide for all n by a single machine only receiving n); certainly such a property will not be polynomially decidable in general; and it will not have useful structural features. Such is the hard property for the dense graph model, constructed in [GGR98] (although [GGR98, Proposition 10.2.3.2] already improves on this by making the property **NPTIME**-decidable). Another improvement, in Goldreich and Trevisan's [GT03, Theorem 1], is an **NPTIME** monotone property; to decide it or to test it, one needs to recognize outputs of a certain pseudorandom generator, making this an **NPTIME** problem not likely to be in **PTIME**. Thus the question stands whether there are even $\Theta\left(n^2\right)$-hard properties which are definitely in **PTIME** while exhibiting most or all of these features. Also, features of properties may be more difficult to establish at $q(n) = o\left(n^2\right)$; specifically, a maximally-hard property is one-sided testable, but in a somewhat meaningless sense: Reading the entire graph meets the query complexity lower bound, and one can thus obviously make a deterministic decision with no error; for $q(n) = o\left(n^2\right)$, a one-sided testability is not at all a trivial matter.

In order to provide hierarchies with these desirable features, we first strengthen the hardness results from [GGR98], by constructing a maximally-hard property which is both **PTIME**-decidable and **PTIME**-testable, in Section 4.2. We use this particular hard property, and the original one of [GGR98, Proposition 10.2.3.1], to establish three hierarchy theorems for the dense graph model, corresponding to three different combinations of the above features:

- **PTIME**-decidability + **PTIME**-testability, in Section 4.5.
- Monotonicity, in Section 4.6.

- **PTIME**-decidability + one-sided testability, in Section 4.7.

A second subtle aspect regards the reductions in the pattern described above for proving hierarchy results. As in Chapter 3, all of these dense model results involve careful use and analysis of graph blowups (see Definition 2.3.6) to relate testing at higher and lower graph orders. Specific to this chapter is the following question: If a graph is far from another graph, or from a property, what guarantee is there that it remains far from it when applying a blowup? The answer is that, in fact, a (balanced) blowup can bring graphs much closer together, even making them identical in some cases; we must therefore prove an appropriate bound on this effect, for different settings in every section, so as to preserve the hardness of properties through blowups. That is perhaps the key to this chapter's dense model results.

4.2 Hard properties decidable and testable in PTIME

Several hierarchy results in this chapter involve hard properties decidable in polynomial time (as per Definition 2.3.2): The result regarding generic functions in Section 4.3, and two of the three dense graph model results, in Section 4.5 and Section 4.7). As our construction of the maximally hard-to-test graph property uses the maximally hard-to-test Boolean function property, we state and establish the existence of both of them together through a single argument:

Theorem 4.1. *There exist a* **PTIME***-decidable property* Π *of generic Boolean functions, and a constant* $\varepsilon_{4.1} > 0$*, such that any* ε*-test for* Π *with* $\varepsilon \leq \varepsilon_{4.1}$ *must make* $\Omega(n)$ *queries, i.e. query at least a constant fraction of the function values.*

Theorem 4.2. *There exist a* **PTIME***-decidable property* Π *of dense graphs, and a constant* $\varepsilon_{4.2} > 0$*, such that for any sufficiently large* n*, any* ε*-test for* Π *with* $\varepsilon \leq \varepsilon_{4.2}$ *must make at least* $c_{4.2} \cdot \binom{n}{2} = \Omega\big(n^2\big)$ *queries, i.e. query at least a constant fraction of the potential edges.*

4.2.1 The difficulties deciding hard-to-test properties in [GGR98]

Let us recap the two-step construction of a hard graph property (of query complexity $\Omega\big(n^2\big)$) in [GGR98, Proposition 10.2.3.1]:

- First, a certain small sample space is shown to yield a hard property of Boolean functions: The sample space is small enough to be sparse, so that a random function is far from it; the sample space also exhibits strong *pseudorandomness*, in that its projection on any (small) constant fraction of the coordinates is close to a projection of a uniformly-sampled random function. Thus a test making at most this many queries cannot tell apart functions sampled uniformly from $\{0,1\}^n$

from functions sampled from the small sample space, while it is necessary for it to usually reject the former and accept the latter.

- Next, the domain of the boolean functions is mapped to the set of (unordered) pairs of graph vertices, and the set of functions is made closed under graph isomorphism (i.e., permutations of the vertices), by adding all isomorphic images of the constituent (labeled) graphs. The result is a graph property, with the original boolean function values corresponding to adjacency matrix entries. The parameters are such, that even though the resulting property may be contain much as $n!$ times as many graphs as the property of Boolean functions, it is still sparse within the set of all possible graphs; a random graph is still far from it; and it still has the strong pseudorandomness with respect to projections — so the indistinguishability is maintained.

There are two difficulties, one in each of the steps of construction, which make the resulting property hard to test in **PTIME**:

- The small sample space used in the first step is in **NPTIME** (that is, one can decide membership in it with an **NPTIME** machine), but it is not clear whether it is in **PTIME**.
  Overcoming this difficulty: Instead of the small sample space used in [GGR98], we shall use another adequate pseudorandom space, the membership in which is decidable in **PTIME**.

- One can easily determine whether a given (labeled) graph is a permutation of a (labeled) graph in the small sample space — using a short witness, being the permutation function (i.e., this can be determined in **NPTIME**). But it is not clear whether this can be done in **PTIME**, without the witness.
  Overcoming this difficulty: We augment the graphs constructed using the Boolean functions, so that after applying an isomorphism (permuting the vertices), the original index of each vertex can be efficiently recovered. Thus the final class can be recognized in **PTIME** by reversing the isomorphism, reconstructing the Boolean function and determining whether it is in the sample space.

4.2.2 The alternative construction

We wish to use a sample space of graphs, the membership in which is efficiently decidable, such that constant-size fractions of it do not reveal enough to make a decision about the entire graph. To this end we begin with such a sample space for binary strings, rather than graphs, which is d-wise independent, i.e. its distribution projected onto any d coordinates is uniform — for $d = \Omega(n)$. The existence of such a space is a long-established result due to Alon, Babai and Itai:

Proposition ([ABI86])**.** *There exists a global constant $\alpha_{\mathrm{ABI}} > 0$ and a linear code, explicitly constructible by a* **PTIME** *Turing machine given n as input, which maps strings of length $n/1000$ to strings of length n, such that every $\alpha_{\mathrm{ABI}} n$ positions in a codeword are linearly independent (and consequently, any assignment to them can be extended to an equal, positive, number of codewords).*

Such a code is constructed using a parity-check matrix spanning a $0.999n$-dimensional vector space (the "dual code"), in which every vector has Hamming weight at least $\alpha_{\mathrm{ABI}} n$. The space of codewords will be our sample space, and the parity-check matrix can be used to efficiently decide membership in the code.

This result in itself is already sufficient for a construction proving Theorem 4.1; but as it will be undertaken as a part of the construction and proof regarding dense graph properties, we shall not set down the proof at this point. Instead, we move from functions to graphs. Consider the same code for $N = \binom{n}{2}$, and fix some efficiently-computable well-ordering $\preceq$ on the set $\{\{i,j\} \mid 1 \leq i,j \leq n\}$.

Definition 4.2.1. For a sequence $s = (s_1, \ldots, s_N) \in \{0,1\}^N$, we define $G_s = ([n], E_s)$, *the graph corresponding to* s where $\{i,j\} \in E_s$ whenever the $\{i,j\}^{\text{th}}$ bit of s, by the order $\preceq$, is 1.

If s is a codeword, G_s is said to be a *codeword* graph. Obviously, as long as a graph is labeled, it can be decided in **PTIME** whether it is a codeword graph or not.

The set of labeled codeword graphs is not in general closed under isomorphism, and does not therefore constitute a graph property. As was done in the [GGR98, Proposition 10.2.3.1] construction, we wish to close the set under isomorphism — but first we must augment the graphs so as to be able to easily recover their original labels. Specifically,

Definition 4.2.2. For a graph $G = ([m], E_s)$ of order m, the (1 mod 4)*-separating augmentation of* G is the graph $G' = ([4m+1], E'_s)$, obtained by adding a $(3m+1)$-vertex labeled clique to G, and connecting every vertex $j \in V$ with the first j vertices of the clique, i.e.

$$E'_s = E_s \cup \{\{u,v\} \mid m+1 \leq u,v \leq 4m+1\} \cup \{\{j, m+\ell\} \mid j \in [m] \wedge \ell \in [j]\}$$

we similarly define the (2 mod 4), (3 mod 4) and (0 mod 4) separating augmentations, in which the large clique is of size $3m+2$, $3m+3$ and $3m+4$ respectively.

The three additional variants of the separating augmentation are defined so that augmented graphs will not be constrained to have a specific order modulo 4 (order $4n+1$ in the basic definition). In most of our analysis below we shall ignore the additional variants, implicitly using the same argument for them as well.

The hard property Π. Our hard property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ shall constitute, at every order $n \geq 5$, the set of isomorphic images of separating augmentations of graphs of order $\lfloor (n-1)/4 \rfloor$, with the original graph G_s having undergone an (n mod 4)-separating augmentation.

Lemma 4.2.3. Π *is decidable in* **PTIME**.

Proof. Consider some $n = 4m + i$, for $i \in \{1, 2, 3, 4\}$. Given a graph of order n, which is the result of a separating augmentation, we note that the vertices originally in the $(3m + i)$-clique are distinguishable from the rest, as their degree is at least $3m$, while the degree of vertices from the pre-augmented graph is at most $m - 1$ to other vertices from the pre-augmented graph, and at most m to vertices in the clique, or $2m - 1$ in total. Having separated the clique and the original vertices, the original index of each original vertex is equal to the number of its neighbors in the clique. We can thus efficiently reconstruct the (single) original order-m graph corresponding to any separating augmentation (or determine that our input is not such an augmentation). Having reconstructed the smaller graph, our earlier arguments implies we can decide in **PTIME** whether the string s corresponding to E_s is a codeword or not. □

To complete our analysis, we shall use Yao's method to demonstrate that Π cannot be tested using $o(n^2)$ queries. Fix some sufficiently large n, let $m = \lfloor (n-1)/4 \rfloor$ and $i = ((n-1) \bmod 4) + 1$, and consider two distributions:

$\mathcal{G}_n$: A uniform distribution over the augmentations of codeword graphs of order m (i.e. over Π_n), and

$\mathcal{R}_n$: A uniform distribution over the augmentations of all graphs of order m.

Note that any n beyond some threshold value can be chosen, as our construction allows for augmentations from any sufficiently large order m to any orders $4m + 1, \ldots, 4m + 4$.

Lemma 4.2.4. *If two graphs G_1, G_2 of order m are δ-far from each other, then (pairs of isomorphic images of) their separating augmentations to order $n = 4m + i$ are $(\delta/32 - O(1/m))$-far from each other.*

Proof. In this proof, as in a few additional ones in this chapter, it will be easier for us to bound distances by accounting for *two-tuple* discrepancies with respect to a bijection between graphs rather than the edge discrepancies, i.e. for every discrepant edge $\{u, v\}$ as per the above, we count both (u, v) and (v, u); this allows us to separate the counts for each vertex in G. As there are no self-loops in our graphs, the number of tuple discrepancies is exactly double the number of edge discrepancies.

Let G'_1, G'_2 denote the augmentations of the two far graphs. Clearly, a bijection which maps (the copy of) G_1 to (the copy of) G_2 exhibits at least

$$2\delta\binom{m}{2} = \frac{2\delta}{16}\left(\binom{4m+i}{2} - \frac{(12+8i)m + i(i-1)}{2}\right) = \frac{2\delta}{16}\binom{4m+i}{2} \cdot \left(1 - O\left(\frac{1}{m}\right)\right)$$

discrepancies.

Now suppose that some vertex v of G_1 in G'_1 is mapped to a vertex of the large clique in G'_2. v is connected to at most $2m-1$ vertices in G'_1 (m in the large clique and $m-1$ in G_1), while the large clique vertex in G'_2 is connected to at least $3m$ vertices. This mapping of v therefore incurs more than m discrepancies of the form (v,u). We conclude that by mapping G_1 vertices to G'_2 large-clique vertices, one can reduce the number of discrepancies no more than by a factor of $4+O(1/m)$. Thus any bijection between G'_1 and G'_2 has at least $\frac{2\delta}{64}\binom{4m+i}{2}\cdot(1-O(1/m))$ discrepancies, and G'_1 is $(\delta/32-O(1/m))$-far from G'_2. □

Lemma 4.2.5. *The probability of a graph sampled from $\mathcal{R}_n$ being $\delta_{4.2.5}=0.4/64$-close to a graph in Π_n is $o(1)$.*

Proof. Let $\mathcal{R}'_m$ denote the uniform distribution over all graphs of order m; A sample from $\mathcal{R}_n$ can be obtained by applying an augmentation to a sample from $\mathcal{R}'_m$.

Now, Π_n is the set of augmentations of codeword graphs; by Lemma 4.2.4, if a graph sampled from $\mathcal{R}_n$ is 0.4/64-close to a graph in Π_n, then its pre-augmentation graph (that is, its corresponding graph from $\mathcal{R}'_m$) is at least 0.4-close to a codeword graph (for sufficiently large n). It thus suffices to prove that the probability of a graph sampled from $\mathcal{R}'_m$ being 0.4-close to a codeword graph is $o(1)$.

Indeed, this follows from the fact that the codeword graphs are a sparse set: Each codeword graph has at most $(m)! = 2^{O(m\log(m))}$ (labeled) isomorphic images. The sample space size (the number of codeword graphs) is $2^{0.001\binom{m}{2}}$, so the number of their isomorphic images is $2^{(0.001+o(1))\binom{m}{2}}$. There are $\sum_{k=0}^{0.4\binom{m}{2}}\binom{\binom{m}{2}}{k}$ graphs which are 0.4-close to a specific codeword graph (corresponding to the possible choices of $k\le 0.4\binom{m}{2}$ edges to add or remove); and it holds that $\sum_{k=0}^{0.4\binom{m}{2}}\binom{\binom{m}{2}}{k} \le 2^{H_\mathrm{b}(0.4)\cdot\binom{m}{2}} = o\Big(2^{0.971\cdot\binom{m}{2}}\Big) = 2^{(0.972+o(1))\cdot\binom{m}{2}}$ — where $H_\mathrm{b}(\cdot)$ denotes the binary entropy function, which satisfies $H_\mathrm{b}(0.4)<0.971$. Thus, for a sufficiently large n, the total number of order-m graphs which are 0.4-close to the set of codewords is under $2^{0.973\cdot\binom{m}{2}}$; since $\mathcal{R}_n$ is uniformly distributed over all $2^{\binom{m}{2}}$ labeled graphs of order m, the claim follows. □

Lemma 4.2.6. *Let M be a probabilistic oracle machine, whose number of queries is at most $d=\alpha_{\mathrm{ABI}}\binom{m}{2} > \big(\frac{1}{16}\alpha_{\mathrm{ABI}}-o(1)\big)\binom{n}{2}$. It holds that $\mathbf{Pr}\big[M^{\mathcal{R}_n}=1\big]=\mathbf{Pr}\big[M^{\mathcal{G}_n}=1\big]$.*

Proof. We establish the claim using two reductions — to distributions over graphs of order m, then to distributions over strings.

Let $\mathcal{G}'_m$ denote the uniform distribution over (labeled) codeword graphs of order m, and let $\mathcal{R}'_m$ denote the uniform distribution over all graphs of order m. Both distributions $\mathcal{R}_n$ and $\mathcal{G}_n$ are obtained by applying the same augmentation to samples from G'_m and $\mathcal{R}'_m$ respectively; and the result of each query to an augmented graph depends on one or no edges of the original order-n graph. It therefore suffices to prove the claim assuming queries are made to the original order-m graphs rather than their

augmentations or the isomorphic images thereof — that is, it suffices to prove that one cannot distinguish between $\mathcal{R}'_m$ and $\mathcal{G}'_m$.

Now, the result of a query of a potential edge $\{i,j\}$ in the edge set E_s of a basic graph is the $\{i,j\}^{\text{th}}$ bit of the string s corresponding to G_s. $\mathcal{G}'_m$ corresponds, therefore, to a uniform sample from the d-wise independent space of length-$\binom{m}{2}$ strings, and $\mathcal{R}'_m$ corresponds to a uniform sample of a string of this length. Thus the claim reduces to asserting that using d queries, one cannot distinguish between strings sampled from the d-wise independent sample space and from a uniform distribution, respectively. For non-adaptive tests, this is the definition of the d-wise independence; but adaptivity does not offer an advantage, since for any choice of up to d queries already made, and for any sequence of results for these queries, the conditional distributions for their completion into d query result are the same (and uniform) regardless of the choice of edges to query. A rigorous treatment of this transition from a non-adaptive to an adaptive bound may be found in [Fis04, Section 8]. □

Proof of Theorem 4.2. Our constructed property Π is decidable in **PTIME**, as established by Lemma 4.2.3. Now, set $\varepsilon_{4.2} = \delta_{4.2.5}$ and $c_{4.2} = \frac{1}{17}\alpha_{\text{ABI}}$, and let n be sufficiently large for Lemma 4.2.6 to hold with $d = \frac{1}{17}\alpha_{\text{ABI}}\binom{n}{2}$. An ε-test for Π accepts with probability at least $2/3$ a graph sampled from $\mathcal{G}_n$. By Lemma 4.2.6, if the test makes less than $c_{4.2}\binom{n}{2}$ queries, it will accept a graph sampled from $\mathcal{R}_n$ with the same probability. By Lemma 4.2.5, with probability $1 - o(1)$, a graph from $\mathcal{R}_n$ is $\delta_{4.2.5} = \varepsilon_{4.2}$-far from Π, so the probability of the test accepting graphs in $\mathcal{R}_n$ which are $\varepsilon_{4.2}$-far from Π is at least $2/3 - o(1)$. Thus for every sufficiently large n there exists a specific graph which is $\varepsilon_{4.2} > \varepsilon$-far from Π_n, and is accepted with probability over $1/2$ — a contradiction.□

Proof Sketch for Theorem 4.1. The proof uses a subset of the arguments above — one need not construct anything from strings or functions in the d-wise independent sample space, so the membership decision is clearly in **PTIME**; also, the sample space is itself sparse enough so that a random Boolean function is $\varepsilon_{4.1}$-far from it with high probability. One can thus construct appropriate indistinguishable distributions as for the case of graphs.

4.3 A hierarchy of generic function properties

In the generic function testing model, the objects tested are functions from $[n]$ to a finite domain; as the elements of the tested functions' domain are not interchangeable as in the case of graphs, one can think of such functions as strings. Our construction will only require Boolean functions (or binary strings).

Definition 4.3.1. The *absolute distance* between two functions $f, g : [n] \to \{0,1\}$ is the number of elements of $[n]$ on which they differ. The *(relative) distance* $\text{dist}\big(f,g\big)$ between f and g is the absolute distance normalized by a factor of $1/n$.

The definition of a *property* and of *satisfying* a property or being ε-*far* from satisfying it are the same as in the dense graph model (except that the classes are of functions, and the distances are as defined above). An ε-*test* for a property of Boolean functions is also defined as for the dense model, except that a test's oracle access is to a generic function f, with a query being an index $i \in [n]$ and a reply being the value of $f(i)$. A test may, alternatively, receive ε together with n as a parameter, so a single algorithm is used for all values of ε.

Theorem 4.3. *There exists a constant $\varepsilon_{4.3} > 0$, such that for $q : \mathbb{N} \to \mathbb{N}$ with an infinite image (i.e. $\limsup_{n\to\infty} q(n) = \infty$), there exists a property Π of Boolean functions that is testable with one-sided error using $q(n)+O(1/\varepsilon)$ queries and running in time polynomial in its number of queries, but not ε-testable with $o(q(n))$ queries, even with two-sided error, for $\varepsilon \leq \varepsilon_{4.3}$. Furthermore, if $q(n)$ is computable from n in $\mathrm{poly}(n)$ time, then the property is* **PTIME***-decidable, and if it is computable in $\mathrm{poly}(q(n))$ time, then the property has a test whose running time is polynomial in its number of queries.*

Note. We assume that the test is given n as input in binary representation rather than in unary, otherwise the computation of $q(n)$ can only be polynomial in $q(n)$ if n is polynomial in $q(n)$.

4.3.1 Property construction

For the rest of this section, fix $q(\cdot)$.

Observation 4.3.2. We may assume, without loss of generality , that $q(n) \leq n/2$, as otherwise we could replace $q(n)$ with $q'(n) = \lfloor \max(q(n)/2, 1) \rfloor$, and Theorem 4.3 would yield a property with the same features but a different constant.

The complexity-q property. Let $\Pi' = \bigcup_{m\in\mathbb{N}} \Pi'_m$ be a property of Boolean functions which requires $\Theta(n)$ queries to test, and is **PTIME**-decidable as a property of strings; Theorem 4.1 guarantees that such properties exist.

Now, let m, n be such that $m = q(n)$. For some $f' \in {\Pi'}_m$, consider the function $f(i) = f'(1 + (i - 1 \bmod q(n))) = f'(1 + (i - 1 \bmod m))$. The domain of f is $[n]$; and it consists of $\lfloor n/q(n) \rfloor$ duplicate copies of f' with perhaps another final incomplete copy. With this construction in mind, our property of query complexity $q(n)$ shall be $\Pi^q = \bigcup_{n\in\mathbb{N}} {\Pi^q}_n$, with ${\Pi^q}_n$ consisting of the functions f constructed for all f' in ${\Pi'}_m$, for $m = q(n)$.

Observation 4.3.3. If $q(n)$ is computable from n in $\mathrm{poly}(n)$ time, then Π^q is decidable in **PTIME**: To decide whether f over domain $[n]$ is in Π^q, one computes $q(n)$, determines whether its corresponding f' is in ${\Pi'}_m$ (for $m = q(n)$), and checks whether $f(i) = f(i+m)$ for every $i \leq n - m$.

4.3.2 Lower and upper query complexity bounds

Lemma 4.3.4. *Testing Π'_m is $(f, 1, h)$-reducible to testing Π^q, for $f(\varepsilon) = \varepsilon/2$ and the partial function $h(m) = \min\{i \in \mathbb{N} \mid q(i) = m\}$ (defined at orders n for which the set is non-empty).*

Proof. Let $m \in \mathbb{N}$ be such that $n = h(m)$ is defined, and consider some Boolean function f' over domain $[m]$. One can construct the function f, corresponding to f', over domain $[n]$ as in the construction of Π^q; if $f' \in \Pi'$, then $f \in \Pi^q$, and if f' is ε-far from Π', one must change an ε-fraction of every complete copy of f' in f to obtain a function in Π', so over all of values f, one must change at least a $\frac{\lfloor n/m \rfloor \cdot m}{n} \cdot \varepsilon$ fraction to obtain $\lfloor n/m \rfloor$ duplicate copies of a function in Π'. (It may be the case that less or no changes are necessary to the incomplete copy of f'). As $q(n) \leq n/2$, this fraction is at least $\varepsilon/2$. Given oracle access to f', one can simulate an oracle to f, making one query to f' so as to answer a single query made to f. This meets the requirements of Definition 2.4.1.□

The lower bound follows as a corollary of the reduction above, when setting $\varepsilon_{4.3} = f(\varepsilon_{4.1}) = \varepsilon_{4.1}/2$:

Lemma 4.3.5. *For $\varepsilon \leq \varepsilon_{4.3}$, any ε-test for Π^q makes $\Omega(q(n))$ queries.*

Proof. By Lemma 4.3.4 above, testing Π'_m is $(f, 1, h)$-reducible to testing Π^q, with a linear $f(\varepsilon)$ and the partial function $h(m) = \min\{i \in \mathbb{N} \mid q(i) = m\}$, defined for m in the image of $q(\cdot)$; since $q(\cdot)$'s image is infinite, the $\Omega(n)$ lower bound for testing Π' when $\varepsilon \leq \varepsilon_{4.1}$ implies (by Lemma 2.4.2) a lower bound of $\Omega(q(n))$ on the number of queries required to test Π^q when $\varepsilon \leq f(\varepsilon_{4.1}) = \varepsilon_{4.3}$ □

For the upper bound, we present a straightforward test for Π^q, listed as Algorithm 4.1.

Algorithm 4.1 A test for Π^q

1: **for** $\Theta(1/\varepsilon)$ times **do**
2: Uniformly sample $j \in [q(n)]$ and $r \in [\lceil n/q(n) \rceil - 1]$.
3: If $r \cdot q(n) + j \leq n$ and $f(r \cdot q(n) + j) \neq f(j)$, reject.
4: **end for**
5: Query all of $f(1), \ldots, f(q(n))$.
6: Reconstruct the function $f' : [q(n)] \to \{0, 1\}$ s.t. $f'(i) = f(i)$.
7: Deterministically decide whether $f' \in \Pi'$ and answer accordingly.

Lemma 4.3.6. *Algorithm 4.1 is a non-adaptive one-sided-error test for Π^q making $q(n) + O(1/\varepsilon)$ queries. Furthermore, if $q(n)$ is computable in $\mathrm{poly}(q(n))$ time, then the test's running time is polynomial in the number of queries.*

Proof. The number of queries of Algorithm 4.1 is clearly as stated. The first part of the test has running time linear in $1/\varepsilon$; then come the steps dependent on n, being

the computation of $q(n)$, then the querying of $q(n)$ values and the decision whether f' is in Π'_m. The test's running time is spent computing $q(n)$, determining whether the first $q(n)$ values are in Π', and additional work taking time linear in $q(n)$ (ignoring an addition of $O(1/\varepsilon)$). As Π' is decidable in **PTIME**, the decision takes time polynomial in $q(n)$. Thus if $q(n)$ is computable in $\text{poly}(q(n))$, the test's overall running time is polynomial in $q(n) + O(1/\varepsilon)$, being its number of queries.

As for completeness and soundness: If $f \in \Pi$, then by definition it is a repetition of some f' in Π' and will therefore be accepted. On the other hand, if f is accepted with probability at least $2/3$, then the f' constructed by the test is necessarily in Π', and f must be ε-close to a repetition of f' — as otherwise the first phase of the test would reject with probability at least $\varepsilon/2$ at every iteration (again, since $q(n) \leq n/2$), and thus with probability at least $2/3$ over all iterations. Thus if f is ε-far from Π^q then the test accepts it with probability lower than $1/3$. □

Theorem 4.3 is now proven by a combination of the query complexity lower bound of Lemma 4.3.5 and the upper bound established through the valid test in Lemma 4.3.6, and Observation 4.3.3 regarding the **PTIME**-decidability.

4.4 An aside: A hierarchy of bounded-degree graph properties

This section regards testing *bounded-degree* graphs, in which any single vertex is connected to at most d vertices: $|\Gamma(v)| \leq d$. Respecting this bound, E is represented in this model by a function:

Definition 4.4.1. For a graph $G = (V, E)$ with maximum degree d, an *edge function* is a function $g_G : V \times [d] \to V \cup \{\perp\}$ such that $g(v, i) = u \in V$ if u is the i^{th} neighbor of v (by some arbitrary order) and $g(v, i) = \perp$ if v has less than i neighbors.

The neighbors of v in G are $\{g(v, 1), \ldots, g(v, \deg(v))\}$.

Definition 4.4.2. The *absolute distance* between two bounded-degree graphs G, H of order n is the minimum distance between pairs of edge functions g_G, g_H representing them, which is in turn the number of values one has to modify in g_G to get g_H. The *(relative) distance* $\text{dist}\big(G, H\big)$ between G and H is the absolute distance between them normalized by a factor of $1/dn$.

Note. Unless one wishes to test bounded-degree digraphs, it must be the case that whenever $u = g(v, i)$ for some i, $v = g(u, j)$ for some j; and there are in fact only at most $dn/2$ edges. Any modification of the edge function must respect this constraint.

The definition of a *property* and of *satisfying* a property or being *ε-far* from satisfying it are the same as in the dense graph model, except for the different normalization of

the distance. A *property test* in the bounded-degree model for a graph property Π is defined as in the dense model, except that its oracle access is to a graph edge function g_G, and its queries are to specific values of this function ("what is the index of the i^{th} neighbor of v?").

Note. As in the dense model, one could alter the definition to remove the artificial dependence of tests on n through the need to use vertex indices, but since this chapter is concerned with tests that depend on n, we shall not explore this here.

Theorem 4.4. *In the bounded-degree model, there exist constants $d \in \mathbb{N}$ and $\varepsilon_{4.4} > 0$, such that for every $q : \mathbb{N} \to \mathbb{N}$ with an infinite image, there exists a (downward) monotone property of degree-d-bounded graphs that is testable with one-sided error using $O(q(n)/\varepsilon)$ queries, but not ε-testable using $o(q(n))$ queries, even allowing two-sided error, for any $\varepsilon \leq \varepsilon_{4.4}$. Particularly, the property of degree-d-bounded graphs being 3-colorable, while having connected components of size at most $q(n)$, is such a property.*

To establish any hierarchy theorem for the bounded-degree model in the common pattern of this chapter (and similarly to Theorem 4.3), we need a property known to be maximally hard. As implied in the statement of Theorem 4.4 above, for bounded-degree graphs this shall be the property of being 3-colorable, which is also monotone, and exhibits some additional features which we shall make use of in the proof. It is known to be hard, by a result of Bodganov, Obata and Trevisan:

Theorem ([BOT02, Theorem 2])**.** *There exist constants $\varepsilon_{3\text{-}COL}$ and d, such that any ε-test of d-bounded graphs for being 3-colorable makes $\Omega(n)$ queries, even when allowed to have two-sided error, for any $\varepsilon \leq \varepsilon_{3\text{-}COL}$.*

From the remainder of this section, let us fix d to be as guaranteed by [BOT02, Theorem 2], and fix also $q(\cdot)$, assuming without loss of generality that $q(n) \leq n/2$ (see Observation 4.3.2 for the justification; here we would be dividing q by $2d$ rather than 2 to obtain the inequality).

The complexity-q property Let Π' be the property of being 3-colorable, and denote $\Pi' = \bigcup_{m\in\mathbb{N}} \Pi'_m$. Our property is $\Pi^q = \bigcup_{n\in\mathbb{N}} \Pi^q{}_n$, with $\Pi^q{}_n$ consisting of all graphs made up of connected components of size at most $q(n)$, which are all 3-colorable, i.e. every connected component itself satisfies Π'.

4.4.1 Lower and upper query complexity bounds

Lemma 4.4.3. *Testing 3-colorability is $(f, 1, h)$-reducible to testing Π^q for $f(\varepsilon) = \varepsilon/2$ and the partial function $h(n) = \min\{i \in \mathbb{N} \mid q(i) = n\}$ (at orders n for which $h(n)$ is defined).*

Proof. For $n = \min\{i \in \mathbb{N} \mid q(i) = m\}$, and given a graph G' of order m and degree at most d, consider a graph G consisting of $t = \lfloor n/m \rfloor$ disjoint disconnected copies of G', and an additional $n \bmod m$ isolated vertices. If $G' \in \Pi'$, then $G \in \Pi^q$. For the case of G' being ε-far from Π', we note that due to the downward monotonicity of Π^q, the distance of G from Π^q is the number of edges one must remove to achieve a graph in Π^q (i.e. there is no benefit in adding edges).

We also note, that graphs in Π^q are themselves 3-colorable (being disjoint unions of 3-colorable graphs), so the edges removed must make G a 3-colorable graph. This requires in particular making every induced subgraph of G 3-colorable, including its (previously) connected components. We conclude that the minimum number of edge removals necessary is exactly the number of edges whose removal is required to make each connected component 3-colorable in itself. To make one of the connected components 3-colorable, we must remove at least an ε-fraction of its edges, and the overall fraction of edges to be removed is at least $\frac{t \cdot m}{n}\varepsilon$. As by assumption $q(n) \leq n/2$, this fraction is at least $\varepsilon/2$, so G is $\varepsilon/2$-far from Π^q.

Finally, given oracle access to an edge function of G', one can simulate an oracle to an edge function of G: For some $v' \in V'$, being the j^{th} copy of some $v \in V$, its i^{th} neighbor will be the j^{th} copy of the i^{th} neighbor of v.

This meets the requirements of Definition 2.4.1. □

The lower bound follows as a corollary of the reduction above, when setting $\varepsilon_{4.4} = f(\varepsilon_{3\text{-COL}}) = \varepsilon_{3\text{-COL}}/2$:

Lemma 4.4.4. *For $\varepsilon \leq \varepsilon_{4.4}$, any ε-test for Π^q makes $\Omega(q(n))$ queries.*

Proof. By Lemma 4.3.4 above, testing Π'_m is $(f, 1, h)$-reducible to testing Π^q, with a linear $f(\varepsilon)$ and $h(m) = \min\{i \in \mathbb{N} \mid q(i) = m\}$, for m in the image of $q(\cdot)$; since $q(\cdot)$'s image is infinite, the $\Omega(n)$ lower bound for testing Π' when $\varepsilon \leq \varepsilon_{3\text{-COL}}$ implies a lower bound of $\Omega(q(n))$ on the number of queries required to test Π^q when $\varepsilon \leq f(\varepsilon_{3\text{-COL}}) = \varepsilon_{4.3}$. □

For the upper bound, we present a test for Π^q, listed as Algorithm 4.1. As in the case of generic functions, the test is quite straightforward.

Lemma 4.4.5. *Algorithm 4.2 is a one-sided-error test for Π^q making $q(n) \cdot O(1/\varepsilon)$ queries.*

Proof. For every iteration of the main loop of Algorithm 4.2, we make at most $d \cdot q(n)$ queries before either deciding that the component is too large or querying the entire component; thus the number of queries of Algorithm 4.2 is as stated. If $G \in \Pi^q$, by definition it consists of components of size at most $q(n)$ which are in Π', and will therefore not be rejected. On the other hand, if G is accepted with probability at least $2/3$, then it must be the case that at most an ε-fraction of the vertices lie in components

Algorithm 4.2 A test for Π^q

1: **for** $\Theta(1/\varepsilon)$ times: **do**
2: Uniformly sample $v \in V(G)$.
3: **while** The neighbors of vertices known to be in the connected component of v have not all been queried **do**
4: Query another unknown neighbor of a vertex in the connected component of v.
5: If v's component is now known to contain at least $q(n)+1$ vertices, reject.
6: **end while**
7: If v's connected component (now fully explored) is not in Π', reject.
8: **end for**
9: accept.

which the test would reject for being too large not in Π'. These can all be made into isolated vertices, by removing at most $\varepsilon \cdot dn$ edges, i.e. G in this case is close to satisfying Π^q. Thus if G is ε-far from Π^q, then the test accepts it with probability lower than $1/3$. □

Theorem 4.4 is now proven by a combination of the query complexity lower bound of Lemma 4.4.4 and the upper bound established through the valid test in Lemma 4.4.5.

Note. Theorem 4.4 also holds for higher values of d beyond the minimum guaranteed by [BOT02, Theorem 2], by the same construction, when adjusting $\varepsilon_{4.4}$ to account for the higher number of possible edges.

Is it possible, as an improvement over the Theorem 4.4, to construct the property with query complexity $\Theta(q(n))$ such that the test's dependence on ε is additive rather than multiplicative? i.e. obtain a test with query complexity $\Theta(q(n) + 1/\varepsilon)$ as is the case in Theorem 4.3? One can alter the construction above so that the graph is made up of 'marked' components, all being copies of the same 3-colorable graph, and use it with some out-of-component gadgets for marking a graph over the copies of the same vertex in the various components. This super-imposed graph could be used to ensure that every pair of copies of two vertices is connected in all components, or in none of them. However, one can't use this sparse graph to check arbitrary pairs of components, as there would be $\Omega(n/q(n))$ components and one would need a walk of length at least $\Omega(\log(n/q(n)))$ in the super-imposed graph to reach all of them, even if the graph were an expander or a balanced tree. One would also need to ensure the super-imposed graph to be appropriate — but this in itself may not be an easy task: [GR02] presents a lower bound of $\Omega(\sqrt{n})$ for testing an order-n bounded-degree graph for the property of having a certain degree of expansion.

4.5 A hierarchy of PTIME-testable properties

In this and the next two sections we return to the dense model for property testing, specifically to dense graphs, proving three hierarchy theorems for three possible combi-

nations of features of properties of arbitrary complexity. The first of them, presented in this section, regards properties which are **PTIME**-decidable and **PTIME**-testable, using the hard property constructed in Theorem 4.2, carrying its **PTIME** decidability to the properties themselves at any query complexity and the testability essentially also to the optimal tests for these properties.

Definition 4.5.1. A function $q : \mathbb{N} \to \mathbb{N}$ is said to be a *reasonable* query complexity function for dense graphs if $q(n) \leq \binom{n}{2}$, and the image of $q(\cdot)$ is infinite, that is, $\limsup_{n\to\infty} q(n) = \infty$.

Theorem 4.5. *There exists a constant $\varepsilon_{4.5} > 0$, such that for every reasonable $q(\cdot)$ (in the sense of Definition 4.5.1), there exists a property of dense graphs that is testable with two-sided error using $O\big(q(n)/\varepsilon^2\big)$ queries, but not ε-testable with $o(q(n))$ queries, even allowing two-sided error, for $\varepsilon \leq \varepsilon_{4.5}$. Furthermore, if $q(n)$ is computable from n in $\mathrm{poly}(n)$ time, then the property is* **PTIME***-decidable, and if it is computable in $\mathrm{poly}(q(n))$ time, then the property has a test whose running time is polynomial in its number of queries.*

4.5.1 Property construction

Vertex dispersal

Our property Π^q will consist of copies, or rather, blow-ups, of graphs from a maximally-hard property, similarly to the proof of Theorem 4.3. However, in order for us to be able to tell vertices apart from each other after their having been blown up, we would like the neighborhoods of different vertices in pre-blown-up graphs to be "substantially different" from each other:

Definition 4.5.2. Let $\alpha > 0$. A graph G of order n is said to be *α-dispersed* if, for every two different vertices $u, v \in V(G)$, their neighbor relations disagree on at least αn elements. In other words, $|(\Gamma(v) \setminus \Gamma(u)) \cup (\Gamma(u) \setminus \Gamma(v))| \geq \alpha n$. A set of graphs is said to be *dispersed* if there exists a single $\alpha > 0$ such that all graphs constituting the set are α-dispersed.

Note. This notion of dispersion has nothing to do with the notion of dispersers as relaxed randomness extractors (as surveyed in Shaltiel's [Sha04]).

We begin with the maximally-hard graph property of Theorem 4.2, denoted here $\Pi' = \bigcup_{n\in\mathbb{N}} \Pi'_n$, which has query complexity $\Theta\big(n^2\big)$, and is also **PTIME**-decidable. We now augment the graphs from Π', so as to make them dispersed:

Definition 4.5.3. Let $G = (V, E)$ be a (labeled) graph of order n. Supposing for ease of notation that $V = [n]$, the *dispersing augmentation* of G consists of:

1. Setting $n' = 2^{\lceil \log_2(2n+1) \rceil} \in [2n + 1, 4n]$.

2. Adding $n'/2$ isolated vertices to the graph *before* the original vertices, making the vertex set $[n'/2+n]$ (with vertices $n'/2+1,\ldots,n'/2+n$ being the vertices of G).
3. Adding an n'-clique to the graph, making the vertex set $[n+3n'/2]$.
4. For every vertex $i \in [n'/2+n]$ (the original and isolated vertices), adding an edge between vertex i and vertex $n'/2+n+j$ (the j^{th} vertex of the large clique) whenever the inner product of $i-1$ with $j-1$, when viewed as $\log_2(n')$-bit strings, is 1 rather than 0.

Notes.
– A dispersing augmentation is a different operation than the separating augmentation used in Subsection 4.2.2; however, since dispersing augmentations are the only ones used in this section, we refer to them throughout the rest of the section merely as *augmentations*.
– Graphs of different orders have dispersing augmentations of different orders; this is the reason why we do not simply augment to size $2n'$. Additional motivation for the specifics of the definition can be found in their use below.

Pre-blowup construction and analysis

The dispersed pre-blowup property We set Π'' to constitute all isomorphic images of dispersing augmentations of graphs from Π'.

The dispersed set Π'' is a graph property — albeit empty for (infinitely) many graph orders. Each labeled graph in Π'' consists of a large clique, a smaller graph from Π' with some additional isolated vertices and a "Hadamard-like" bipartite graph between them. Π'' is not the final property we shall be testing, but in order to complete our construction we must establish several of its features:

Lemma 4.5.4. *Π'' is $1/8$-dispersed, and the minimum degree of graphs in Π''_n is higher than $n/4$.*

Proof. Let $G'' \in \Pi''$, and let n be such that G'' is an isomorphic image of the augmentation of a graph of order n. Showing that the neighborhoods of every two vertices in G'' differ by at least $n'/4$ vertices establishes the claim regarding dispersion.

Let us consider the neighborhoods of pairs of vertices based on vertices' membership in the large clique:

- For the case of one vertex outside the large clique and another from the large clique, the large-clique vertex is connected to $n'-1$ other vertices in the large clique, while the other vertex of the pair is connected to exactly $n'/2$ of the large clique vertices (by construction of the Hadamard-like bipartite graph), so the two neighborhoods differ on at least $n'/2-1 \geq n'/4$ vertices.
- For two vertices i_1, i_2 outside the large clique, their neighborhoods differ with respect to each large clique vertex j such that the inner product of j with $i_1 \oplus i_2$ is 1, i.e. the neighborhoods differ on $n'/2$ of the large-clique vertices.

- For pairs of large clique vertices, an analogous claim as for in-clique pairs could have held for them, had there been exactly n' non-large-clique vertices; this is not the case, but let us think of them as having 'lost' a single difference of their neighborhoods for every isolated vertex 'missing' from the augmentation: At most $n' - (n + n'/2) = (n'/2 - n) \leq (n'/2 - n'/4) = n'/4$; thus the neighborhoods of pairs of large-clique vertices differ on at least $n'/2 - n'/4 = n'/4$ vertices.

Also, any vertex, either in or out of the large clique, is connected to at least $n'/2$ vertices of the large clique, hence the minimum degree is $n'/2 \geq |V(G'')|/4$. □

We now wish to show that the dispersing augmentation preserves distances, but before doing so we require the following simple result:

Lemma 4.5.5. *Let G_1, G_2 be two graphs of order n at distance ε. If one adds $n' - n$ full-degree vertices, or alternatively $n' - n$ isolated vertices, to each of the graphs, their distance becomes exactly $\varepsilon \cdot \binom{n}{2} \Big/ \binom{n'}{2} > \varepsilon(n/n')^2$. Specifically, there are optimal bijections between the augmented graphs in which G_1 is mapped to G_2.*

Proof. We prove for the case of isolated vertices; the case of full-degree vertices is similar.

Clearly, by taking the optimal bijection between G_1 and G_2 and expanding it into a bijection between the additional isolated vertices, one obtains a bijection with $\varepsilon\binom{n}{2}$ discrepancies.

In the other direction, denote by G_1' and G_2' the graphs with the isolated vertices added. Suppose a bijection $\phi : V(G_1') \to V(G_2')$ maps some vertex $v \in V(G_1)$ to an isolated vertex; ϕ must have some isolated vertex of G_1' mapped to some $u \in V(G_2)$. If one remaps v to u, the number of edge discrepancies does not increase: $|\Gamma(v)| + |\Gamma(u)|$ discrepancies from mapping v and the isolated vertex respectively are avoided, and the number of discrepancies added is the size of the symmetric difference of the two neighborhoods, $\Gamma(v)$ and $\Gamma(u)$, which is at most the number avoided. We thus conclude that there is an optimal ϕ which maps $V(G_1)$ to $V(G_2)$, so the minimum number of discrepancies cannot be less than $\varepsilon\binom{n}{2}$. □

Lemma 4.5.6. *If G is ε-far from Π', then the dispersing augmentation of G is $\varepsilon/250$-far from Π''.*

Proof. Let n be the order of G, and let G' be its dispersing augmentation, having $n'' = n + \frac{3}{2}n' = n + 2^{\lceil \log_2(2n+1) \rceil}$ vertices. To be in $\Pi''_{n''}$, a graph must be an augmentation of an order-n graph in Π'; specifically, it must have an induced copy of some $\widetilde{G} \in \Pi'$, an appropriate number of isolated vertices and a large clique. If G is ε-far from Π, and one only modifies the copy of G in G', with the isolated vertices and the large clique vertices keeping their respective roles, then the fraction of G' which must be modified to obtain a graph in Π'' is $\varepsilon\binom{n}{2}/\binom{n''}{2} > \varepsilon\binom{n}{2}/\binom{7n}{2} > \varepsilon/50$.

However, one might also have vertices of G' outside of the copy of G take the role of G vertices. To bound the effect of such mappings, let us consider the tuple discrepancies (rather than edge discrepancies) in a bijection from G' to some graph in Π''.

First suppose that large-clique vertices are mapped to non-large-clique vertices (G or isolated vertices), and vice-versa. Specifically, let v in G or an isolated vertex be mapped to a large-clique vertex v', with some large-clique vertex u mapped to a $\widetilde{G}$-vertex or isolated vertex u'.

Had the augmentation had n' non-large-clique vertices, each large-clique vertex would be connected to $n'/2$ of them by the Hadamard-like graph. The augmentation has $n'/2 + n$ non-large-clique vertices, so each large-clique vertex is connected to at least n of them. A large-clique vertex therefore has degree at least $(n' - 1) + n$, while a G, $\widetilde{G}$ or isolated vertex has degree at most $n'/2 + (n - 1)$; thus u must have at least $n'/2 + 1$ vertices removed and v must have at least $n'/2 + 1$ vertices added, i.e. v, v', u and u' contribute at least $n' + 2$ discrepancies. If one were to map v to u and v' to u', the number of discrepancies would have been at most $4n'$ (discrepancies in the Hadamard-like bipartite graph) plus $2(n - 1)$ (discrepancies within G), less than $5n'$ in total. Thus by altering the mapping as just described, the number of discrepancies increases by a factor of 5 at most; overall, with the same maximum increase factor in discrepancies, one can avoid any mapping of large-clique vertices to G or isolated vertices.

Now, if large-clique vertices are only mapped to large-clique vertices, the discrepancies under the mapping can be divided into discrepancies within the Hadamard-like bipartite graph, and discrepancies within the set of G and isolated vertices. By Lemma 4.5.5, this latter number of discrepancies is no less than $2\varepsilon\binom{n}{2}$ (as the isolated vertices were added to two graphs at distance ε), so the overall number of discrepancies is at least this much. Having increased the overall number of discrepancies by at most a factor of 5 by enforcing no large-clique vertex to be mapped to a non-large-click vertex, we conclude that an unconstrained bijection has at least $\frac{1}{5} \cdot 2\varepsilon\binom{n}{2} > \frac{2}{250}\varepsilon\binom{7n}{2}$ tuple discrepancies, so the distance of G' from Π'' is at least $\varepsilon/250$. □

Lemma 4.5.7. *Testing Π' is $(f(\varepsilon), 1, h''(n))$-reducible to testing Π'', for $f(\varepsilon) = \varepsilon/250$ and $h''(n) = \frac{3}{2} \cdot 2^{\lceil \log_2(2n+1) \rceil} + n \leq 7n$.*

Proof. For a graph G, we simulate an oracle to the corresponding augmentation G' of G, with $\frac{3}{2}n' + n = \frac{3}{2} \cdot 2^{\lceil \log(2n+1) \rceil} + n = h''(n)$ vertices. Queries involving the the large clique, the Hadamard-like bipartite graph or the $n'/2$ isolated vertices of G' can be answered without making any queries to G, based on vertex indices only; queries regarding pairs of vertices from the pre-augmented G are simply passed to the oracle to G. If G satisfies Π', then its augmentation satisfies Π'', by definition; the other direction is Lemma 4.5.6: If G is ε-far from Π', then G' is $\varepsilon/250 = f(\varepsilon)$-far from Π''. □

Lemma 4.5.8. Π'' *is* **PTIME***-decidable.*

Proof. Suppose a graph G' of order n' is in Π''. G' is a dispersing augmentation of some G in Π' of order n (satisfying $n' = \frac{3}{2} \cdot 2^{\lceil \log(2n+1) \rceil} + n$, an equation from which n can be calculated efficiently). We can easily tell apart the large clique vertices in G' from all other vertices, as their degree is at least $2^{\lceil \log(2n+1) \rceil} + \lfloor n/2 \rfloor$ while the degree of other vertices is at most $2^{\lceil \log(2n+1) \rceil - 1} + n - 1 < 2^{\lceil \log(2n+1) \rceil}$. We can also tell apart the isolated vertices added to G', as G itself has no isolated vertices (and we can ignore the edges to the large clique, which we have set apart). Now, having located the (isomorphic) copy of G in G', we can ensure in **PTIME** that it is indeed in Π'', as Π'' itself is **PTIME**-decidable. It remains to ensure that the bipartite graph between the large clique and the other vertices is Hadamard-like, as in the definition of the augmentation.

This would be perfectly immediate with $2^{\lceil \log(2n+1) \rceil}$ vertices both inside and outside the large clique — in that case the bipartite graph's adjacency matrix is a full square Hadamard matrix, and vertex indices can be permuted according to a permutation of the $\lceil \log(2n+1) \rceil$ bits in the index of a vertex, with the permutation resulting again in a Hadamard matrix. In this case, one can verify the adjacency matrix by locating $\lceil \log(2n+1) \rceil$ large clique vertices, with the i^{th} refining the partition induced by the previous ones from 2^{i-1} into 2^i cells of equal size. One then ensures that the vertices outside this 'basis' each have a neighborhood which is the exclusive-or operation of a unique subset of the basis vertex neighborhoods.

In our case, we can also successively locate appropriate large-clique vertices, but more careful accounting is necessary. We proceed from the most to the least significant bits — a concept which has meaning in our graph, as one cannot simply permute the vertices outside the clique (there are not enough of them, and isolated and G vertices are not interchangeable). Thus one begins by finding a clique vertex disconnected from all isolated vertices and connected to all G vertices. It is necessarily the first vertex of the large clique (according to the original labeling).

One can now limit the focus to the isolated vertices, and successively locate vertices splitting the existing partition cells of the isolated vertices into halves, choosing such a vertex with the minimum number of neighbors in G. At some point one gets the same number of non-neighbors and neighbors in G for the splitting large-clique vertex, and from this point on the order of bits within indices of G and isolated vertices is immaterial, since these bits may be permuted without requiring that vertices be missing from G'. The successive choice of large-clique vertices for the 'basis' can from that point on ignore the balance of G neighbors and non-neighbors and proceed as in the simple case of the full Hadamard matrix. Eventually an appropriate 'basis' of $\lceil \log(2n+1) \rceil$ vertices is obtained and the other large-clique vertices can be verified to each uniquely correspond to an index in $\left[2^{\lceil \log(2n+1) \rceil}\right]$. Thus $G' \in \Pi''$ will be accepted by a **PTIME** computation.

Conversely, if the algorithm sketched out above accepts G', it has appropriately

partitioned it into large-clique, isolated and G vertices; verified that G is in Π''; and that the bipartite graph to the large clique is valid. This implies G' is in Π''. □

Blowup construction and analysis

Definition 4.5.9. For $n \in \mathbb{N}$, the Π^q *pre-blowup order* for n, denoted $m(n,q)$ or m for short, is the highest integer up to and including $\lfloor\sqrt{q(n)}\rfloor$, for which Π''_m is not empty, or 1 if there is no such integer.

Observation 4.5.10. $m(n,q)$ is no lower than $\frac{1}{2}\sqrt{q(n)} - 1$ (comparing $\frac{3}{2}n' + n$ for consecutive values of n). Consequently, $\binom{m(n,q)}{2} = \Theta(q(n))$.

The complexity-q property. $\Pi^q = \bigcup_{n\in\mathbb{N}} {\Pi^q}_n$ is the property for which ${\Pi^q}_n$ contains all (isomorphic images) of blowups of graphs in ${\Pi''}_{m(n,q)}$ to order n. In other words, a graph in ${\Pi^q}_n$ has m clusters of size either $\lfloor n/m \rfloor$ or $\lceil n/m \rceil$, with complete bipartite graphs of edges between these cluster pairs corresponding to edges of a graph in ${\Pi''}_{m(n,q)}$.

Lemma 4.5.11. *If $q(n)$ is computable from n in* poly(n) *time, then Π^q is* **PTIME***-decidable.*

Proof. By Lemma 4.5.4, Π'' is dispersed; specifically, the neighborhoods of each vertex in a graph in Π'' are distinct. This holds after blowup; that is, the neighborhoods of vertices from different clusters are distinct. Thus, given an order-n graph G, one can cluster it; ensure that G is a blowup, with $m(n,q)$ clusters, all of size $\lfloor n/m \rfloor$ or $\lceil n/m \rceil$; reconstruct a pre-blown-up graph G'' of order m; and determine whether $G'' \in \Pi''$. Since Π'' is **PTIME**-decidable (as per Lemma 4.5.8 above), $q(n) \leq \binom{n}{2}$, and the other tasks can all be carried out in time polynomial in n, we conclude that the total time necessary is polynomial in n.

4.5.2 A query complexity lower bound for the constructed property

As in the case of generic functions, this lower bound uses a reduction from testing Π at order n to testing Π' at a lower order — as we've augmented and blown up graphs from Π'. However, unlike replications of generic function, blowups — even exactly-balanced ones — do not necessarily preserve the distance between graphs; see [GKNR10] for an example due to Arie Matsliah, of a constant factor decrease in distance when blowing up corresponding graphs from two families. When blowups are not exactly balanced, distance can even be nullified:

Example 4.5.12. Let G_1 have one isolated vertex and a 2-path on two vertices, and let G_2 have two isolated vertices and two connected vertices; the distance between G_1 and G_2 is $1/6$ — removing one edge of the 2-path makes G_1 into G_2. Now consider a blowup of these order-4 graphs to order 5: In G_1 the isolated vertex is replicated,

while in G_2, one of the connected vertices is replicated. The (unlabeled) result of both blowups is a graph consisting of two isolated vertices and a 2-path on three vertices; thus the distance has dropped from 1/6 to 0.

Fortunately, while the distance may decrease, this change can be bounded when one of the graphs is dispersed, even for the more problematic case of n not dividing n', where the blowup cannot be exactly-balanced:

Lemma 4.5.13. *There exists a global constant $c_{4.5.13} > 0$, such that for every n, ε, α and every pair of (unlabeled) graphs (G_1, G_2) of order n, with G_1 being α-dispersed, the following holds: If G_1 and G_2 are ε-far from each other, then any pair of (balanced) blowups of G_1 and G_2 to order n', are at a distance of at least $c_{4.5.13} \cdot \alpha \cdot \varepsilon$ from each other.*

Note. In the case of exactly-balanced blowups, an even stronger result of Oleg Pikhurko, published independently of our work, holds: The distance between the blowups is no lower than a third of the original distance, regardless of their dispersal ([Pik10, Lemma 14]).

Proof of Lemma 4.5.13. Roughly, we argue that the dispersal feature of G_1 makes it approximately optimal to map complete clusters of one graph to complete clusters of the other to the extent possible, rather than splitting clusters of the first graph among several clusters of the other graph.

Let us label the vertices of both graphs, so that we may denote $V(G_1) = V(G_2) = [n]$ (this induces a labeling of the blowup clusters). Let G_1' and G_2' denote the respective blowups of the two graphs. Let $t = \lfloor n'/n \rfloor$; the clusters in G_1' and G_2' all have either t or $t+1$ vertices. The *(relative) weight* of a cluster of s vertices in a graph G, denoted ρ, is the fraction $s/|V(G)|$; the relative weight of a pair of clusters is the product of their weights.

Consider a bijection $\pi' : V(G_1') \to V(G_2')$ which minimizes the number of discrepancies; in the context of this proof we will be counting the tuple (ordered pair) discrepancies of π' rather than the edge discrepancies.

If the blowups were exactly-balanced (that is, with every cluster having exactly $t = n'/n$ vertices), and every cluster of G_1' were mapped by π' to a cluster of G_2' (of the same size), one could construct a corresponding map $\pi : V(G_1) \to V(G_2)$, with t^{-2} as many discrepancies as π'; and since G_1 and G_2 are ε-far, this would imply that π' has at least $t^2 \cdot 2\varepsilon\binom{n}{2} \approx 2\varepsilon\binom{nt}{2}$ discrepancies (the distance can't be preserved exactly, since the fraction of (v, v) tuples is smaller in larger graphs; if one normalizes distances by n^2, then one gets an equality here).

Now, we refer to a cluster of G_2 of weight ρ as *pegged* if it has more than $\frac{1}{2}\rho$ of its weight, over a half, in vertices which π' maps from a single cluster of G_1, and as *unpegged* otherwise. (e.g. note that a cluster of size 2 is pegged if and only if both

of its vertices are mapped from the same cluster). Also, let ε' be such that there are exactly $2\varepsilon' {n'}^2$ discrepancies under π'.

We first show that the total weight P of unpegged clusters in G_2 is bounded as a function of ε'.

Indeed, consider an unpegged cluster in G_2' with weight ρ. This cluster must have vertices mapped to it from at least two clusters of G_1. Order all of its source clusters by decreasing number of vertices mapped to the G_2' cluster, breaking ties arbitrarily, and start taking a union of them from the first on. At some point the union of clusters contains between $\rho/3$ and $2\rho/3$ of the vertices. Now match arbitrarily as many as possible of these vertices to vertices from the remaining source clusters. The result is a set of at least $\rho n'/3$ disjoint pairs of vertices mapped to the unpegged G_2' cluster, each two coming from different clusters of G_1'. Every such pair contributes at least $\alpha t n$ discrepancies to the total count: The two vertices' neighborhoods disagree on αn vertices in G_1, and at least $\alpha \cdot tn$ in the blowup G_1' (possibly significantly more). When mapped to the same cluster in G_2', they must be made to have the same neighborhood; regardless of which neighborhood this is, for every disagreement, one of the two vertices must have an edge removed or added.

The set of all unpegged clusters, having total weight P, contributes, therefore, $\frac{\mathrm{P}n'}{3} \cdot \alpha t n = \frac{\alpha \mathrm{P}}{3} n'(tn) \geq \frac{\alpha \mathrm{P}}{3} \frac{(n')^2}{2} = \frac{\alpha \mathrm{P}}{6}(n')^2$ discrepancies. If $\mathrm{P} > 12\varepsilon'/\alpha$, this exceeds $2\varepsilon' {n'}^2$, the total number of discrepancies — an impossibility.

For a pegged G_2'-cluster, consider the G_1'-cluster being the source of the majority of its vertices. Can such a G_1'-cluster be the source for two separate G_2' clusters? Indeed, it can, for $t = 1$ — a 2-vertex G_1' cluster pegging two 1-vertex G_2' clusters. However, by the pigeonhole principle, for each such G_1' cluster, there must exist some 2-vertex G_2' cluster whose two vertices come from different G_1' clusters, i.e. an unpegged G_2' cluster of the same weight. Thus the total weight of two-pegging G_1' -clusters is no higher than the weight of unpegged G_2' clusters, $12\varepsilon'/\alpha$; and the weight of the G_2' clusters they peg is at most $24\varepsilon'/\alpha$. Let us refer to these clusters as *jointly-pegged* and to the rest of the pegged clusters as *singularly-pegged*.

Now, consider a bijection π between the vertices of G_1 and G_2, such that for every singularly-pegged G_2'-cluster i_2 and its source G_1'-cluster i_1, π maps vertex i_1 to vertex i_2, i.e. π "agrees with the majority mapping" of π' for singularly-pegged clusters; the rest of π is set arbitrarily. This definition is consistent, as the singular pegging ensures that our definition does not make two constraints on the mapping of a single G_1 vertex.

We note that discrepancies under π, of (i, j) with $(\pi(i), \pi(j))$, can be 'charged' to discrepancies under π', if the G_2' clusters corresponding to $\pi(i)$ and $\pi(j)$ are singularly-pegged: If $(\pi(i), \pi(j))$ is discrepant with its source tuple (i, j), then the majority of vertices in G_2' cluster $\pi(i)$ form a discrepant tuple with vertices from the majority in the cluster $\pi(j)$ — because their sources under π' are vertices in clusters i and j respectively.

Now, As G_1 and G_2 are ε-far, π must have at least $2\varepsilon\binom{n}{2} > \varepsilon n^2$ discrepancies (counting tuples rather than sets). Thus the total number of vertex pairs in pairs of G_2'

clusters, whose corresponding G_2 vertex pairs are discrepant under π, is at least $\frac{1}{4}\varepsilon n'^2$ (as for a given π discrepancy, the product of the corresponding cluster weights is at least $(t/n')^2 > (1/2n)^2$). Up to three quarters of these pairs ($\frac{3}{16}\varepsilon n'^2$) may have at least one vertex not originating in the pegging G'_1 cluster, and are therefore not known to be discrepant; also, less than $4 \cdot \mathrm{P}n'^2$ of these pairs may involve vertices from unpegged or doubly-pegged clusters. The remaining pairs must be discrepant under π' as well. We thus arrive at an inequality relating ε and ε':

$$\begin{aligned}\frac{1}{4}\varepsilon n'^2 - \frac{3}{16}\varepsilon n'^2 - 4 \cdot \mathrm{P}n'^2 &< 2\varepsilon' n'^2 \\ \frac{1}{16}\varepsilon - 48\frac{\varepsilon'}{\alpha} &< 2\varepsilon' \\ \varepsilon &< 32\varepsilon' + 768\frac{\varepsilon'}{\alpha}\end{aligned}$$

This implies $\varepsilon' > \alpha\varepsilon/800$. The fraction of discrepant edges (rather than tuples) under π' is therefore $\varepsilon' n'^2 \Big/ \binom{n'}{2} > \varepsilon'/2$ and the claim follows for $c_{4.5.13} = 1/1600$. □

Lemma 4.5.14. *Testing Π'' is $(f, 1, h')$-reducible to testing Π^q, for $f(\varepsilon) = (c_{4.5.13}/8) \cdot \varepsilon$ and $h'(n) = \min\{i \in \mathbb{N} \mid \lfloor\sqrt{q(i)}\rfloor = n\}$ (at orders n for which $h'(n)$ is defined; in which case $n = m(h'(n), q)$).*

Proof. Even for orders n for which $h'(n)$ is defined, we only consider those orders for which ${\Pi''}_n$ is non-empty (as otherwise, a trivial test for Π'' will simply reject).

Given a graph G of appropriate order n, we apply the blowup to order $h'(n)$ as in the construction of Π^q, obtaining a graph G'. By our construction, If G satisfies Π'', G' satisfies Π^q. In the other direction, consider a graph G which is ε-far from Π''. Since G is ε-far from every individual graph in Π'', and by Lemma 4.5.4, it is also $1/8$-dispersed, we may apply Lemma 4.5.13, and conclude that the blowup G' of G is $(c_{4.5.13}/8) \cdot \varepsilon$-far from the blowup of every graph in Π'', that is, far from every graph in ${\Pi^q}_{h'(n)}$, and hence this far from Π^q as a property.

Also, Given oracle access to G, one can easily simulate an oracle to G', using at most one query to G for the answer to any query made to G'. □

Lemma 4.5.15. *Testing Π' is $(f(\varepsilon), 1, h(n))$-reducible to testing Π^q, for*

$$\begin{aligned}h(n) &= h'(h''(n)) = \min\left\{ i \in \mathbb{N} \;\middle|\; \left\lfloor\sqrt{q(i)}\right\rfloor = \frac{3}{2} \cdot 2^{\lceil \log_2(2n+1)\rceil} + n \right\} \\ f(\varepsilon) &= c_{4.5.13}/2000 \cdot \varepsilon\end{aligned}$$

(at orders n for which $h'(n)$ is defined).

Proof. The reduction is a combination of the reductions from testing Π' to testing Π'' (as per Lemma 4.5.7, and from testing Π'' to testing Π (as per Lemma 4.5.14). □

We can now prove the lower bound, setting $\varepsilon_{4.5} = c_{4.5.13}/2000 \cdot \varepsilon_{4.1}$:

Lemma 4.5.16. *For $\varepsilon \leq \varepsilon_{4.5}$, any ε-test for Π^q makes $\Omega(q(n))$ queries.*

Proof. By Lemma 4.5.15 above, testing Π'_m is $(f, 1, h)$-reducible to testing Π^q, with $f(\varepsilon) = c_{4.5.13}/2000 \cdot \varepsilon$ and $h(n) = \min\left\{ i \in \mathbb{N} \;\middle|\; \lfloor\sqrt{q(i)}\rfloor = \frac{3}{2} \cdot 2^{\lceil \log_2(2n+1)\rceil} + n \right\}$. $h(n)$ has an infinite image, and $f(\varepsilon)$ is continuous and contains the interval $(0, c_{4.5.13}/2000)$. Also, the lower bound for testing Π' with $\varepsilon \leq \varepsilon_{4.1}$ is $q'(n) = \Omega\big(n^2\big)$ queries. Finally, $q(n)$ has is infinite range, thus so does $h(n)$. We now apply Lemma 2.4.2, and obtain a lower bound of $\Omega\big(\big(h^{-1}(n)\big)^2\big)$ on the number of queries required to test Π^q with $\varepsilon \leq f(\varepsilon_{4.5}) = \varepsilon_{4.5}$. The proof is completed by noting that when $h^{-1}(n)$ is defined, its value is $\Theta\big(\sqrt{q(n)}\big)$ (see also Observation 4.5.10). □

4.5.3 A test for the constructed property

An adaptive, two-sided error test for Π^q is listed as Algorithm 4.3. For clarity of analysis, the test makes the assumption that n/m is an integer, in which case the graphs in Π are exactly-balanced blowups, with no need to account for the small difference in cluster sizes; we later argue that this assumption can be foregone.

Algorithm 4.3 A test for Π^q

1: $\varepsilon' \leftarrow \varepsilon/5$, $m \leftarrow \lfloor\sqrt{q(n)}\rfloor$.

Phase I: Clustering and representative vertex selection.

2: $S_{\text{sig}} \leftarrow$ uniform sample of $\Theta(\log(m))$ *signature* vertices.
3: $S_{\text{csize}} \leftarrow$ uniform sample of $s' = \Theta\big(m\log(m)/\varepsilon'^2\big)$ *cluster size estimation* vertices.
4: Query all edges between S_{csize} and S_{sig}.
5: Cluster the vertices of S_{csize} using the known part of their neighborhoods.
6: If the number of clusters is not exactly m, reject.
7: If any cluster has size outside the range $(1 \pm \varepsilon')s'/m$ vertices, reject.
8: $R \leftarrow$ An arbitrarily selection of one representative vertex in S_{csize} from each cluster.

Phase II: Representative validation.

9: **for** $\Theta(1/\varepsilon)$ times: **do**
10: Uniformly select a pair of vertices u, v.
11: Cluster u and v using their neighborhoods in S_{sig}.
12: If u or v are in none of the m existing clusters, reject.
13: Let $r_u, r_v \in R$ denote the representative vertices of the two vertices' clusters.
14: If $\{u, v\} \in E$ and $\{r_u, r_v\} \notin E$, or vice-versa, reject.
15: **end for**

Phase III: Checking the pre-blown-up graph.

16: Query the order-m subgraph induced by R.
17: Decide whether the induced subgraph is in Π', and answer accordingly.

Lemma 4.5.17. *Algorithm 4.3 is a valid test for Π^q, making $O(q(n))$ queries. Furthermore, if $q(n)$ is computable in $\text{poly}(q(n))$ time, then the test's running time is polynomial in its number of queries.*

Proof. The number of queries made by Algorithm 4.3 is dominated by the queries of edges between S_{sig} and S_{csize}, and by the querying of the m-vertex subgraph in the final step. The number of queries made there is $O\big(\varepsilon^{-2} \cdot m\log^2(m)\big) + \binom{m}{2}$; as we are interested in the dependence on m, this is $\Theta\big(m^2\big) = \Theta(q(n))$.

As for running time, the potentially time-consuming parts of the test are computing $q(n)$ and deciding whether the order-m subgraph induced by the representatives is in Π'; if the former task can be carried out in $\text{poly}(q(n))$ time, then by Lemma 4.5.11 the latter task will require $\text{poly}(\sqrt{q(n)})$ time, which is specifically $\text{poly}(q(n))$, so this part of the claim holds.

We now turn to the test's completeness. Let $G \in \Pi_n$ be a blowup of $G' \in \Pi''_m$. By Lemma 4.5.4, Π'' is $\frac{1}{8}$-dispersed; thus for every pair of (different) vertices $u, v \in G'$, a uniformly-sampled vertex is located in only one of their neighborhoods, with probability at least $\frac{1}{8}$. For a sample of $\Theta(\log(m))$ vertices uniformly, the probability that the neighborhoods of u and v to have the same intersection with all sampled vertices is less than $\frac{1}{6}\binom{m}{2}^{-1}$. The same is true when u, v are vertices of G, from different clusters, and the signature vertices are sampled from G rather than in G'. Union-bounding over all $\binom{m}{2}$ pairs of G' vertices, we find that with probability at least $1 - \frac{1}{6}$, the signature vertices induce the m clusters in G corresponding to the vertices of G', each of size n/m. Also, the probability that the fraction of sampled validation vertices from a certain cluster is outside the range $(1 \pm \varepsilon') \cdot s'/m$ is less than $\exp\big(\Omega\big(-\varepsilon'^2 s'/m\big)\big) < 1/6m$; thus with probability at least $1 - \frac{1}{6}$, S_{csize} contains $(1 \pm \varepsilon') \cdot s'/m$ vertices from each cluster. Assuming all of the above occurs, Phase I does not reject; Phase II cannot reject since the u and r_u, v and r_v have the same neighborhoods respectively; and in Phase III the test correctly reconstructs the pre-blowup graph G' (regardless of which representatives were chosen) and accepts, as G' is in Π''. Thus the probability of accepting G is at least $2/3$.

It remains to establish the soundness of the test. Let us suppose that an input graph G is accepted with high constant probability (e.g. $1/3$), and show that it cannot be ε-far from Π.

The signature set S_{sig} is said to be a *good* signature if the clustering it induces has at least m clusters each of size $(1 \pm 2\varepsilon')n/m$ (and a *bad* signature otherwise). We first show that the high probability of acceptance implies that G has a good signature, which is sampled as S_{sig}. Assume to the contrary that S_{sig} is bad.

If S_{sig} only induces less than m clusters, then the test must reject in Phase I, so we assume at least m clusters are induced. Suppose some m vertices of S_{sig} originate in m of these induced clusters. Now consider the distribution of the $s' - m$ remaining vertices of S_{csize}. These must contain between $(1-\varepsilon')s'/m - 1$ and $(1+\varepsilon')s'/m - 1$ vertices from each of these induced clusters, for Phase I not to reject G. Specifically, the remainder of S_{csize} must contain this number of vertices from some S_{sig}-induced cluster C, whose size in G is not in the range $(1 \pm 2\varepsilon')n/m$. Now consider the $n - m$ remaining vertices

of G, from which the remainder of S_{csize} is sampled; the fraction of C vertices among these is outside the range $(1 \pm 1.5\varepsilon')(n-m)/m$ (due to our implicit assumption that $n \geq s'$; see Definition 2.1.4.) The probability of the remainder of S_{csize} containing the necessary fraction of vertices from C is therefore $\exp\bigl(-\Omega\bigl((0.5\varepsilon')^2 s'/m\bigr)\bigr) < 1/3$. Thus under our assumption that the signature is bad, Phase I rejects G with probability at least $2/3$ — contradicting our assumption that G is accepted with high probability. It must therefore be the case that G has good signatures, and one of these is sampled as S_{sig}.

Now, a representative set $R = \{r_1, \ldots, r_m\}$ is said to be *well-representing* (with respect to a signature S_{sig}) if all of the following holds:

(i) Each r_i is in a different cluster induced by S_{sig}.

(ii) An ε'-fraction of the vertex pairs u, v of G are such that both u and v are in represented clusters, and are consistent with their representatives with respect to E (that is, $\{u, v\} \in E$ iff $\{r_u, r_v\} \in E$).

(iii) The subgraph of G induced by R is in Π''_m.

If the test does not reject by the end of Phase I, it must have found m clusters induced by S_{sig}, so the set R it obtains obeys requirement (i). If R fails to obey (ii), it will be rejected with probability greater than $2/3$ at Phase II, due to an unrepresented vertex or an inconsistent pair of vertices ; if R fails to obey (iii), it will be rejected at Phase III, deterministically. Thus, under our assumption that the test accepts, there must exist some well-representing set R, with respect to the good signature set, which the test obtains.

Fixing a good signature set S_{sig} and a well-representing set R, let $\{C_1, \ldots, C_m\}$ denote the set of m clusters induced by S_{sig}, and let V_{nc} denote the set of vertices not belonging to any of the m clusters. One can redistribute the excess vertices in each C_i, and the vertices of V_{nc} among the C_i's, so that each C_i becomes of size n/m exactly (at most $2\varepsilon' n/m$ additions or removals in each cluster). One then needs to modify the edges incident on redistributed vertices to match the subgraph induced by R; this requires at most $5\varepsilon'\binom{n}{2}$ changes: Up to $4\varepsilon'\binom{n}{2}$ vertex pairs whose endpoints have been reassigned to a different cluster, plus up to $\varepsilon'\binom{n}{2}$ pairs which had been in disagreement with their representative pair with respect to E. This results in a graph satisfying Π, and as $5\varepsilon' = \varepsilon$, we conclude that G is indeed indeed ε-close to Π under our assumption. The claim follows. □

Note. In the above, the large deviation bounds are applied as though the vertices sampled are independent, while when a set of vertices is sampled without replacements, this is not the case. However, large deviation bounds do apply to samples without repetition from a finite set (in fact, even slightly more tightly than to independent samples). This is established in [Hoe63], in the discussion preceding and following Theorem 4 there. That theorem implies that the same or similar bounds established elsewhere (e.g. [ASE92, Appendix A]), through examining the expectation of the

exponent of the sum of independent bounded variables, also apply to samples without repetition.

Observation 4.5.18. The test in Algorithm 4.3 can be generalized to the case in which n/m is not an integer. The modification required is to allow for any $n \bmod m$ of the clusters to have desired size $\lceil n/m \rceil$ while the others have $\lfloor n/m \rfloor$ (or actually ensure that the validation vertices' intersections with the clusters are of relative sizes between $(1-\varepsilon')\lfloor n/m \rfloor/n$ and $(1+\varepsilon')\lceil n/m \rceil/n$). In the analysis of the test, the "well-representing" sets will be respective of specific choices of $m \bmod n$ larger clusters.

Theorem 4.5 is now proven by combining the lower bound of Lemma 4.5.16 and the upper bound established through the valid test in Lemma 4.5.17, together with Lemma 4.5.11 regarding **PTIME**-decidability (while taking Observation 4.5.18 into account).

4.6 A hierarchy of monotone properties

This section continues Section 4.5, with a second hierarchy theorem for dense graph properties. In this section, instead of focusing on the **PTIME** testability, the additional feature we ensure for properties of arbitrary query complexity is *upwards monotonicity*. The direction of monotonicity is inconsequential, as one notes that an identical result holds for downwards monotone properties by considering the complements of graphs in the upwards-monotone property; we hereon in this section refer to upwards-monotone properties as simply 'monotone'. Unlike the first and third hierarchy results, the construction here does not utilize the **PTIME**-decidable hard-to-test property constructed in Section 4.2, but rather the hard property of [GGR98, Proposition 10.2.3.1], which is generally very hard to decide deterministically, but whose simpler construction better allows us to place other relevant constraints on its constituent graphs.

Theorem 4.6. *There exist a constant $\varepsilon_{4.6} > 0$, such that for every reasonable $q(\cdot)$ (in the sense of Definition 4.5.1), there exists an (upwards) monotone property of dense graphs that is testable with two-sided error using $O\big(q(n)\varepsilon^{-4}\log^2\big(\varepsilon^{-1}\big)\big)$ queries (or $O(q(n))$ if one ignores the dependence on ε), but is not ε-testable using $o(q(n))$ queries, even allowing two-sided error, for $\varepsilon \leq \varepsilon_{4.6}$.*

4.6.1 Property construction

Our construction of a property which is both monotone and hard-to-test will effectively involve the taking of what is at the same time a blowup and a monotone closure of another property. This operation must maintain not only an $\Omega(q)$ lower bound on testing, but also the upper bound, the possibility of testing with $O(q)$ queries. This is a challenge, as the possible addition of edges can 'drown out' much of the structure

of the graph. We shall overcome this difficulty with a combination of two measures: The first is that if "too many" edges have been added relative to the original graph, then we will allow ourselves to always accept, thus limiting the hardness to graphs with average degree in a certain range; the second measure is constraining the graphs to have additional structural features which are robust enough, so that few edge additions do not disrupt them overmuch. This second measure is achieved through the choice of our initial, hard-to-test property.

Revisiting the hard property of Goldreich, Goldwasser and Ron

We wish to begin our construction with a hard property satisfying several additional constraints:

Lemma 4.6.1. *There exists a (not generally monotone) graph property $\Pi' = \bigcup_{n\in\mathbb{N}} \Pi'_n$ with the following features. First, a probabilistic oracle machine making $o\big(n^2\big)$ queries can only distinguish with probability $o(1)$ between a uniformly-sampled graph from Π'_n and a graph sampled from distribution $G\big(n, \frac{1}{2}\big)$. Also, for every δ and sufficiently large n (as a function of δ), every graph $G \in \Pi'_n$ satisfies:*

1. *Every vertex in G has degree $\big(\frac{1}{2} \pm \delta\big)n$.*
2. *For every pair of vertices in G, the union of their neighborhoods contains $\big(\frac{3}{4} \pm \delta\big)n$ vertices.*

Also, every two graphs $G_1, G_2 \in \Pi'_n$ satisfy:

3. *If G_1, G_2 are non-isomorphic, then they are 0.4-far from each other.*
4. *If G_1, G_2 are isomorphic, but their isomorphism fixes less than $0.9n$ of the vertices, then they differ on at least $0.01\binom{n}{2}$ of their edges. In other words, and letting $[n]$ denote the graphs' vertex set, if the isomohrphism $\pi : [n] \to [n]$ is such that $|\{i \in [n] \mid \pi(i) \neq i\}| > 0.1n$, then the identity bijection between G_1 and G_2 induces at least $0.01\binom{n}{2}$ edge discrepancies.*

Finally, in addition to the above, an n-vertex graph, sampled from the $G\big(n, \frac{1}{2}\big)$ distribution (i.e. each vertex pair being an edge with probability $1/2$, independently of the others), is 0.4-far from Π' with probability $1 - o(1)$.

Proof. Let Π' be the property constructed in the proof of [GGR98, Proposition 10.2.3.1], with two slight modifications. Π' is obtained there by sampling $K = 2^{\frac{1}{10}\cdot\binom{n}{2}}$ (labeled) graphs using the $G\big(n, \frac{1}{2}\big)$ distribution, and closing the resulting set to isomorphism by taking all isomorphic images of the sampled labeled graphs. Our first modification will be setting K differently, to $2^{\frac{1}{1000}\cdot\binom{n}{2}}$; the construction remains the same with the alternate K, except that the query complexity lower bound for distinguishing between Π' and a random graph with probability $1/2$ drops from some $c \cdot n^2$ to some $c' \cdot n^2$ for some global constants $0 < c' < c$; see the proofs of [GGR98, Proposition 10.2.3.1] and [GGR98, Proposition 4.1.1] for details. We also note that if $c'n^2$ queries are required to

distinguish between the distributions with probability 1/2, then with $o\big(n^2\big)$ queries, one can only distinguish between them with probability $o(1)$.

We now turn to the degree constraints and our second modification of the property.

The probability of a single graph out of the K failing to satisfy either of the first two degree constraints is at most $O\big(n^2\big) \cdot \exp(-\Omega(\delta n))$ (using standard large-deviation bounds, and for n sufficiently large to ignore the neighborhood size being $n-1$ rather than n). We introduce a second modification to the construction, which is the removal of these unsatisfactory graphs (and their isomorphic images) from Π'; few enough graphs are removed so that the argument regarding distinguishing graphs sampled from Π' and from $G\big(n, \frac{1}{2}\big)$ still holds; and the density of Π' remains very close to the original.

with probability $1-o(1)$, Π' (as a set of labeled graphs) has close to $n!K$ constituent graphs at order n. By a union bound, the probability of a $G\big(n, \frac{1}{2}\big)$ graph being 0.4-close to it is therefore less than this number times the probability of a $G\big(n, \frac{1}{2}\big)$ graph being close to a specific (labeled) graph. This latter probability is equal to the probability of a graph having at most $0.4\binom{n}{2}$ edges (considering these edges as changed edges from the original graph), which is less than $\exp\big(-0.02\binom{n}{2}\big)$; it is therefore the case that a $G\big(n, \frac{1}{2}\big)$ graph is 0.4-far from Π' with probability $1-o(1)$.

A similar argument can be used to establish the third constraint: Fixing graphs $G_1, \ldots, G_{s-1}$ in Π', the graph G_s sampled into Π' (before any removal of graphs) is merely a sample from $G\big(n, \frac{1}{2}\big)$. The probability of this sample being 0.4-close to any of the previous $s-1$ graphs sampled into Π' or their isomorphic images is $o(1/K)$; union-bounding over all K samples, we conclude that the third constraint is indeed met with probability $1-o(1)$.

It remains to establish the fourth constraint in the statement of the lemma. Consider an arbitrary permutation π over $[n]$, fixing less than $0.9n$ vertices. We wish to show that a large enough subset E' of the pairs in G_1 satisfies $E' \cap \pi(E') = \varnothing$; if this is the case, we can use the fact that G_1 is sampled from $G\big(n, \frac{1}{2}\big)$ and conclude that the number of discrepancies of E' by an identity bijection between G_1 and G_2 is close to $\frac{1}{2}E'$ with high probability.

Indeed, let $U = \{i \in [n] \mid \pi(i) \neq i\}$ be the set of unfixed elements, with $|U| = \alpha n$. Let $I \subseteq U$ be a subset of them such that $|I| = \left\lfloor \frac{1}{3}\alpha n \right\rfloor$ and $\pi(I) \cap I = \varnothing$. Such a set exists, as a greedy algorithm can construct it by repeatedly adding another unfixed element $i \in U$, and marking $\pi(i)$ as unacceptable for addition. We now set $E' = \{\{u, v\} \mid u \in I \wedge v \in V \setminus (I \cup \pi(I))\}$. These edges have no endpoint in $\pi(I)$, and are mapped by π to pairs with one endpoint in $\pi(I)$, so that $E' \cap \pi(E') = \varnothing$ as desired. Thus under the identity bijection between G_1 and G_2, every pair in E' is mapped to a pair out of E'. As the edges of G_1 are chosen to exist independently of each other and with probability 1/2, the expected number of discrepancies of the identity bijection is

therefore at least

$$\frac{1}{2}\left|E'\right| = \frac{1}{2}\cdot|I|\cdot(|V|-2|I|) = \frac{1}{2}\cdot\left\lfloor\frac{1}{3}\alpha n\right\rfloor\cdot\left(n-2\left\lfloor\frac{1}{3}\alpha n\right\rfloor\right) = \frac{1}{18}\alpha n^2\cdot(3-2\alpha)-O(n)$$

In the range $0.1 \leq \alpha \leq 1$, the minimum of the first term is achieved at 0.1 and the expression is $\frac{1}{18}\cdot 0.1n^2\cdot(3-0.2)-O(n) > 0.015\binom{n}{2}$ for sufficiently large n. Now, the probability that one specific G_1 and one specific isomorphic image $G_2 = \pi(G_1)$ have less than $0.01\binom{n}{2}$ discrepancies in E' is at most $\exp\left(-2\cdot 0.005\binom{n}{2}\right)$. Union-bounding over all K initial graphs in Π' and all their permutations fixing less than $0.9n$ of the vertices, we conclude that with probability $1-o(1)$, all such pairs indeed have at least $0.01\binom{n}{2}$ discrepancies.

By another union bound using the arguments above, all constraints hold simultaneously with probability $1-o(1)$. □

Note. Regarding the deterministic computational complexity of Π', it may not even be deterministically computable, due to the random sampling. We could replace, in the construction of $\Pi'[n]$, the sampling with an exhaustive search of the first set (by a lexicographic order) of graphs which satisfies the requirements and is of appropriate size; this would ensure computability. Also, it may be possible to devise a construction based on a small **NPTIME**-decidable sample space, as in [GGR98, Proposition 10.2.3.2]; but we do not explore this possibility in this work.

Property construction via approximate monotone blowups

Definition 4.6.2. Let $G=(V,E)$ be a graph of order n and $G'=(V',E')$ a graph of order n'. G' is said to be a *β-threshold approximate monotone blowup* of G if V' can be partitioned into $|V|+1$ clusters of vertices, as follows: The last cluster contains $n' \bmod n$ vertices with full degree $n'-1$; the rest of the clusters are all of size $t=\lfloor n'/n\rfloor$, and each correspond to a vertex $v\in V$; for every $u',v'\in V'$ in clusters corresponding to $u,v\in V$, such that $\{u,v\}\in E$, either $\{u',v'\}\in E'$, or at least one of u' and v' is a *heavy* vertex, having degree at least $n' \bmod n+\beta(tn-1)$ (i.e. at least $\beta(tn-1)$ neighbors within the first $|V|$ clusters).

Note. The condition on edges in E and E' ensures that, ignoring high-degree vertices, clusters in the blowup are in *monotone agreement* with vertices in the original graph: Pairs of clusters in G' corresponding to connected pairs in G have a complete bipartite graph between them; and the bipartite graph between clusters corresponding to a disconnected pair may, or may not, be empty. As we will see below, however, our concern will be with graphs whose overall number of edges is not too high, so that these bipartite graphs cannot 'fill out' overmuch.

Let us now fix some parameters, so as to be able to construct a specific Π^q. As our construction utilizes Lemma 4.6.1, it depends on the value of δ for which we apply that

lemma. We fix $\Delta = 10^{-11}c_{4.5.13}$. This value, the 'leeway' for 'informative' vertex degrees in graphs of our monotone properties, is set low enough to meet certain constraints which come up in in the analysis of the construction, and in the proof of the validity of a test in Subsection 4.6.3 further below.

Now let us fix $q(\cdot)$ for the rest of this section. We assume that $q(n) > q_0 = 100\Delta^{-4}\log^4\left(\Delta^{-1}\right)$. This (non-optimized) bound is necessary for some of our arguments below, as the orders of graphs with which we will be concerned depend on $q(n)$. There is no loss of generality in this assumption regarding q: Recall that q is required to satisfy $q(n) \leq \binom{n}{2}$, and for its image to be infinite. Thus if our additional assumption does not hold, we replace q with $q'(n) = \min\left\{\binom{n}{2}, \max\{q, q_0\}\right\}$. This is still a valid function with respect to the statement of Theorem 4.6, and when plugged in there it yields the same result, albeit with a different threshold distance $\varepsilon_{4.5}$ for hardness. Note that q_0 does not depends on ε.

A complexity-q property. Let $m(n,q) = \left\lfloor\sqrt{q(n)}\right\rfloor$. We set $\Pi^q = \bigcup_{n\in\mathbb{N}} {\Pi^q}_n$ with ${\Pi^q}_n$ containing all graphs $G = (V, E)$ satisfying at least one of the following two conditions:

(C1) The graph has at least $0.5 + 2\Delta \cdot \binom{n}{2}$ edges.
(C2) Each vertex in G has degree at least $(0.5 - \Delta)n$ and G is a 0.52-threshold approximate monotone blowup of a graph in $\Pi'_{m(n,q)}$.

Observation 4.6.3. Π^q is monotone (as each of the two conditions is itself monotone).

4.6.2 A query complexity lower bound for the constructed property

The hard property Π' we use as the base of our construction is proven to be hard, in [GGR98, Proposition 10.2.3.1], using Yao's method, with the far distribution consisting of $G\left(n, \frac{1}{2}\right)$ graphs. As graphs in Π^q are constructed by transforming graphs in Π', our lower bound will use distributions of transformed graphs, in a similar manner to Section 4.2:

$\mathcal{R}_n$: An exactly-balanced blowup of a graph sampled from distribution $G\left(m(n,q), \frac{1}{2}\right)$, to order $n - (n \bmod m)$, to which are added $n \bmod m$ additional vertices of full degree.

$\mathcal{G}_n$: An exactly-balanced blowup of a graph sampled uniformly from Π'_m, to order $n - (n \bmod m)$, to which are added $n \bmod m$ additional vertices of full degree.

(Recall that these distributions are only defined for n sufficiently large so that the constraints on Δ and q may be satisfied, with Lemma 4.6.1 holding.)

Lemma 4.6.4. *The graphs of $\mathcal{G}_n$ are all in ${\Pi^q}_n$.*

Proof. A graph of $\mathcal{G}_n$ is an exactly-balanced blowup of a $\Pi^{q'}$ graph, with additional full-degree vertices — constituting an approximate monotone blowup, regardless of the threshold value: The $n \bmod m$ full-degree vertices count as a separate cluster, and the rest of the graph is in monotone agreement with the $\Pi^{q'}$ graph. Also, the minimum degree of a $\Pi^{q'}$ graph is at least $(0.5-\Delta)m$; the minimum degree of the exactly-balanced blowup is therefore at least $(0.5-\Delta)(n-(n \bmod m))$, and adding full-degree vertices makes the minimum degree no less than $(0.5-\Delta)(n-(n \bmod m))+(n \bmod m) \geq (0.5-\Delta)n$. This meets condition (C2). □

Lemma 4.6.5. *with probability $1-o(1)$, a graph sampled from $\mathcal{R}_n$ is $(0.08 \cdot c_{4.5.13})$-far from the support of $\mathcal{G}_n$.*

Proof. By Lemma 4.6.1, a graph G sampled from distribution $G\left(m, \frac{1}{2}\right)$ is 0.4-far from Π'_m with probability $1-o(1)$. Also, with probability $1-o(1)$ G is 0.4-dispersed. When both these events occur, Lemma 4.5.13 guarantees that any exactly-balanced blowup of G is $(c_{4.5.13} \cdot 0.4 \cdot 0.4)$-far from all exactly-balanced blowups of graphs in Π'_m. Finally, adding $n-(n \bmod m)$ full-degree vertices to all exactly-balanced blowups both of G and of a graph in Π'_m can reduce the distance between them by a factor of no more than $n/(n \bmod m) < 2$ (by Lemma 4.5.5). Thus after applying the entire transformation of the definition of $\mathcal{R}_n$ to G, we have a graph sampled from distribution $\mathcal{R}_n$, which with probability $1-o(1)$ is $0.08 \cdot c_{4.5.13}$-far from the transformed graphs Π'_m graphs; the proof is completed, as these transformed graphs are the support of $\mathcal{G}_n$. □

Lemma 4.6.6. *Let $\delta \leq \Delta/4$, let G' be a graph of order $n' = n-(n \bmod m)$ with maximum degree at most $(0.5+\delta)n'$, and let G be the result of adding $n \bmod m$ vertices to G' with full degree $n-1$. If G is δ-close to ${\Pi^q}_n$, then it is $(63\delta+\Delta)$-close to the support of the $\mathcal{G}_n$ distribution.*

Proof. We first consider values of n which are multiples of m, in which case there are no full-degree vertices added to blowups in the construction of ${\Pi^q}_n$ and no full-degree vertices in G. Also, for this case we only assume $\delta \leq \Delta$.

Let $H \in \Pi^q$ be the satisfying graph closest to G. The number of edges in H is less than $(0.5+\delta)\binom{n}{2} + \delta\binom{n}{2} = (0.5+2\delta) \cdot \binom{n}{2}$ (edges in G plus an upper bound on edges added); H must therefore satisfy condition (C2) rather than (C1) in the definition of Π^q. Let $H' \in \Pi'_m$ be the graph of which H is an approximate monotone blowup, and let H'' be an exactly-balanced blowup of H' to order n. We show that H'' — which is in Π^q, and particularly in the support of $\mathcal{G}_n$ — is close to G.

Denote by V_{heavy} the set of heavy vertices in H, that is, the vertices which have degree greater than $0.52n$ (note that since n is a multiple of m, these do not include any full-degree vertices added to the monotone blowup). Also, for ease of notation we assume $V(G) = V(H) = V(H'') = V$.

The discrepancies between corresponding edges of H and H'' can be attributed to one of two causes: Having a heavy vertex (in V_{heavy}) for an endpoint; or the edge

being in a bipartite graph between clusters of H, whose corresponding H' vertices are disconnected (there is no case of H'' having an edge between non-heavy vertices of H, which is not also present in H — by the definition of an approximate monotone blowup).

Regarding heavy-vertex-endpoint edges, we note that G has no 0.52-heavy vertices, having maximum degree $0.5 + \delta$; thus V_{heavy} contains only as many vertices as is made possible by up to $\delta\binom{n}{2}$ edges it may have in addition to those of G; thus $|V_{\text{heavy}}| \leq \delta\binom{n}{2}/(0.52 - (0.5 + \delta))n < 30\delta(n-1)$. At most $(n-1)$ edge discrepancies between H and H'' edges may be attributed to each such vertex, for a total of less than $60\delta\binom{n}{2}$ edge discrepancies over all of V_{heavy}.

Now suppose we correct all discrepancies in H with H'' involving heavy vertices, i.e. modify the neighborhoods of vertices in V_{heavy} to their values in H''. These vertices now obey the constraints on non-heavy vertices in an approximate monotone blowup — and if we were now to remove all edges existing in H but not in H'', we would get H'' exactly: We would be 'cleaning out' the bipartite graphs corresponding to disconnected H'-vertices. The number of edges we would need to remove would be the difference in the total number of edges between the modified H and H''. Correcting heavy vertices necessarily involves removing more edges to them than are added, so after this correction to H it still has at most $(0.5 + 2\delta) \cdot \binom{n}{2}$ edges. The number of edges in H'' is at least $\frac{1}{2}(0.5 - \Delta)n^2 > (0.5 - \Delta)\binom{n}{2}$ edges, since the average degree of H'', as a graph in Π^q, is at least $(0.5 - \Delta)n$. The number of edges remaining to be removed to make the modified H into H'' is therefore no higher than $(2\delta + \Delta)\binom{n}{2}$.

Altogether, H and H'' are therefore at a distance of no more than $62\delta + \Delta$, so G and H'' are at a distance of no more than $63\delta + \Delta$.

Now let us consider the general case, in which n is not necessarily divisible by m; we wish to reduce it to the case of n divisible by m, for which no full-degree vertices are added.

Let H be as in the previous case: A graph satisfying Π^q at minimum distance from G. H has $n \bmod m$ full-degree vertices; without loss of generality , we may assume that the $n \bmod m$ full-degree vertices in G are mapped to these, and that G' (the exactly-balanced blowup being an induced subgraph of G) is mapped to an induced subgraph H' of H, of order $n' = n - (n \bmod m)$; this assumption is possible by Lemma 4.5.5. H is an approximate monotone blowup of some graph in Π'_m — and H' is also an approximate monotone blowup of the same graph (with the cluster of full-degree vertices being empty). Thus H' satisfies $\Pi^q{}_{n'}$ and, in fact, it is in the support of $G_{n'}$. The distance of G' from H' is $\delta' = \delta\binom{n}{2}\Big/\binom{n'}{2} < 4\delta \leq \Delta$. The argument for the previous case now applies to G' (as even though G' does not meet the requirements of the lemma, it meets the relaxed requirements of the first case discussed above); thus G' is $(63\delta' + \Delta)$-close to $G_{n'}$.

Finally, let H'' denote the $G_{n'}$ graph closest to G'. Adding $n \bmod m$ full-degree vertices to H'' results in a graph which is in G_n, and its distance from G is $(63\delta' + \Delta) \cdot$

$\binom{n'}{2} \Big/ \binom{n}{2} \leq 63\delta + \Delta$. □

Lemma 4.6.7. *with probability* $1 - o(1)$*, a graph sampled from* $\mathcal{R}_n$ *is* $\Delta/4$*-far from* ${\Pi^q}_n$*.*

Proof. Let $\delta = \Delta/4$, and let n be sufficiently large so that with probability $1 - o(1)$, a uniformly sampled graph of order $m(n, q)$ has maximum degree $(0.5 + \delta)m$; its blowup to order $n-(n \bmod m)$ has maximum degree $(0.5 + \delta)(n - (n \bmod m))$. Now, suppose that the graph from $\mathcal{R}_n$ observes this bound (before the addition of the $n \bmod m$ full-degree vertices), and that $\mathcal{R}_n$ is also δ-close to ${\Pi^q}_n$. We may now apply Lemma 4.6.6 to conclude that the graph is $(63\delta + \Delta)$-close to the support of $\mathcal{G}_n$. But since $\Delta = 10^{-5} c_{4.5.13}$, we have $63\delta + \Delta < 0.08 \cdot c_{4.5.13}$. By Lemma 4.6.5, this can be the case only with probability $o(1)$. □

We consequently set $\varepsilon_{4.6} = \Delta/4$.

Lemma 4.6.8. *Any probabilistic oracle machine making* $o\big(m^2\big)$ *queries has probability* $o(1)$ *of distinguishing between inputs from* $\mathcal{R}_n$ *and from* $\mathcal{G}_n$*.*

Proof. The proof is by the same argument as in Lemma 4.2.6: Let $\mathcal{R}'_m$ and $\mathcal{G}'_m$ be the uniform distributions over all graphs of order m and over Π'_m respectively; distributions $\mathcal{R}_n$ and $\mathcal{G}_n$ are obtained by applying the same augmentation to samples from $\mathcal{G}'_m$ and $\mathcal{R}'_m$ respectively. The result of each query to an augmented graph depends on one or no edges of the original order-m graph. It therefore suffices to prove the claim assuming queries are made to the original order-m graphs rather than their augmentations or the isomorphic images thereof; in other words, it suffices to show that the probability of an oracle machine distinguishing between inputs from R'_m and from G'_m, using $o\big(m^2\big)$ queries, is $o(1)$. This is guaranteed by the choice of Π' in Lemma 4.6.1. □

Lemma 4.6.9. *Any* ε*-test for* Π^q*, for* $\varepsilon \leq \varepsilon_{4.6}$*, makes* $\Omega(q(n))$ *queries.*

Proof. Let n be sufficiently large for Lemma 4.6.7 to hold, and let $\varepsilon \leq \varepsilon_{4.6}$. An ε-test for Π accepts with probability at least $2/3$ a graph sampled from $\mathcal{G}_n$. By Lemma 4.6.8, if the test makes $o\big(m^2\big) = o(q(n))$ queries, then for a sufficiently large n it will accept a graph sampled from $\mathcal{R}_n$ with probability at least $2/3 - o(1)$. By Lemma 4.6.7, with probability $1 - o(1)$, a graph from $\mathcal{R}_n$ is $\Delta/4 = \varepsilon_{4.6}$-far from Π, so the probability of the test accepting graphs in $\mathcal{R}_n$ which are $\varepsilon_{4.6}$-far from Π is also at least $2/3 - o(1)$. Thus for every sufficiently large n there exists a specific graph which is $\Delta/4 = \varepsilon_{4.6} > \varepsilon$-far from Π_n, and is accepted with probability over $1/2$ — a contradiction. □

4.6.3 A test for the constructed property

In this subsection we present a test for Π^q. As in Section 4.5, we assume, for the sake of simplicity, that n is an integer multiple of m, in which case graphs in Π^q are

approximate monotone blowups, with no full-degree vertices added to them; we later argue that this assumption can be foregone.

Definition 4.6.10. Let $\Xi = \bigcup_{n\in\mathbb{N}} \Xi_n$ denote the property of graphs satisfying a relaxed version of the conditions of (C2) of Π^q, and failing to satisfy a relaxation of condition (C1). Specifically, a graph G is in Ξ if its minimum degree is at least $(0.5 - 10^5\Delta)n$ (rather than $(0.5 - \Delta)n$ in (C2)), its average degree is lower than $(0.5 + 10^5\Delta)n$ (rather than $(0.5 + 2\Delta)n$, the threshold for (C1)), and it is a 0.52-threshold approximate monotone blowup of a graph in $\Pi'_{m(n,q)}$.

The motivation for this definition is that, when testing a graph satisfying (C2) but not (C1), we hope to reconstruct, by querying $\mathrm{poly}(1/\varepsilon)$ edges, a graph which approximately satisfies (C2) with the above parameters. Before proceeding to present an actual test, we wish to establish the fact that, having reconstructed a Ξ graph, we can also determine the pre-blowup graph of which it is an approximate monotone blowup. This is less than trivial, due to Ξ graphs having some heavy vertices, as well as some edges between clusters which are disconnected in the pre-blowup graph.

Lemma 4.6.11. *Let $G = (V, E)$ be a graph in Ξ_n, for sufficiently large n. There exists a graph $G'' = (V'', E'')$ and a corresponding partition of the non-heavy vertices into m cells (denoted $V'_1, \ldots, V'_m$), so that the following holds:*

1. *V'' has a vertex for each non-empty cell V_i, i.e. $|V''| = |\{i \in [m] \mid V'_i \neq \varnothing\}|$ (and specifically, $|V''| \leq m$).*
2. *G'' is an induced subgraph of some graph in Π'_m.*
3. *Each V'_i contains at most n/m vertices.*
4. *G'' is in monotone agreement with the partition, i.e. for every $\{i, j\} \in E''$ and every $(u, v) \in V'_i \times V'_j$, it holds that $\{u, v\} \in E$.*
5. *At most $0.01m$ sets V_i are empty.*
6. *Neighborhoods of different vertices within the same partition cell agree on all but at most $0.05n$ vertices.*
7. *Neighborhoods of vertices from different partition cells disagree on at least $0.45n$ vertices.*

Proof. Let $V_{\text{heavy}} \subseteq V$ denote the set of heavy vertices of G (those with degree exceeding $0.52n$). Before considering the seven requirements, we first bound from above the number of heavy vertices in G, using the constraint on the minimum and the average degrees: A bound is obtained by assuming that every non-heavy vertex contributes only the minimum degree towards the overall average, and the heavy vertices contribute only $0.52n$ each. In this case a sum over the vertex degrees yields $0.52n \cdot |V_{\text{heavy}}| + \big(0.5 - 10^5\Delta\big)n \cdot (n - |V_{\text{heavy}}|) \leq \big(0.5 + 10^5\Delta\big)n^2$; thus $\big(0.02 + 10^5\Delta\big) \cdot |V_{\text{heavy}}| \leq \big(2 \cdot 10^5\Delta\big)n$; as $(2 \cdot 10^5)/(0.02 + 10^5\Delta) < 10^7$, this implies $|V_{\text{heavy}}| < 10^7\Delta$.

A partition and a graph G'' satisfying the requirements above are the obvious ones: G is a 0.52-threshold approximate monotone blowup of $G' = ([m], E')$ of order m, and G'' is chosen as the subgraph of G' induced by those vertices i with $V_i \neq \varnothing$; this satisfies requirements 1 and 2. The partition chosen is the clustering of V in the approximate monotone blowup, i.e. V'_i is the cluster originating in $i \in G'$, excluding any heavy vertices. This satisfies requirements 3 and 4 (by definition of an approximate monotone blowup). Regarding requirement 5, empty partition cells correspond to clusters with n/m vertices which are all heavy, and there can be at most $|V_{\text{heavy}}|/\frac{n}{m} < 10^7 \Delta m < 0.01m$ of these (for sufficiently large n).

Regarding requirement 6: The neighborhoods of a pair of vertices in the same cluster V_i of the blowup must agree on at least those edges mandated by G', whose endpoints are not heavy; i, as a vertex of G', has degree no lower than $(0.5 - \Delta)m$, so there are $(0.5 - \Delta)n$ edges which both vertices must have, minus up to $|V_{\text{heavy}}|$ edges to heavy vertices which are not constrained to be present: At least $(0.5 - (10^7 + 1)\Delta)n$. On top of these, every one of the two vertices can have at most $0.52n - (0.5 - (10^7 + 1)\Delta)n = 0.02n + (10^7 + 1)\Delta n < 0.021n$ additional neighbors so as not exceed the maximum degree of a non-heavy vertex. Thus the two neighborhoods can differ by at most $2 \cdot 0.021n < 0.05n$ of their neighbors.

Regarding requirement 7: If the blowup had been exactly-balanced rather than monotone, that is, had G not had any heavy vertices, and had G contained only those edges corresponding to edges in G', a pair of vertices in different clusters V'_i and V'_j would each have at least $(0.5 - \Delta)n$ neighbors, of which at least $(0.25 - \Delta)n$ were shared with the other vertex and at least $(0.25 - \Delta)n$ not shared. Thus G' mandates a $(0.5 - 2\Delta)$ fraction of difference between the neighborhoods. As argued above, the heavy vertices and the leeway with respect to the degree of non-heavy vertices can alter the number of vertices in disagreement by at most $2 \cdot 0.021n$ for each vertex. Thus the difference between the neighborhoods is at least $(0.5 - 2\Delta)n - 2 \cdot 0.021n > 0.45n$.

Lemma 4.6.12. *Let G, G' be as in Lemma 4.6.11; the graph G'' and the partition $V'_1, \ldots, V'_m$ guaranteed to exist by Lemma 4.6.11 are unique up to isomorphism (relabeling), and for a given $V'_1, \ldots, V'_m$, the labeling of G'' is unique.*

Proof. Consider an arbitrary graph and partition of $V \setminus V_{\text{heavy}}$ which satisfy all the requirements of Lemma 4.6.11. Now, two vertices from the same G'-cluster cannot be assigned different cells V_i, V_j — as such two vertices have highly different neighborhoods by requirement 7 (of Lemma 4.6.11), and their placement together will violate requirement 6 (their neighborhoods will not be consistent enough). For a similar reason, a pair of vertices from different clusters cannot be assigned the same cell in the partition — their neighborhoods will differ while required to agree. It must therefore be the case that the partition is exactly $V_1, \ldots, V_m$, up to a reordering. Let us assume without loss

of generality this is exactly the partition (without reordering). It now remains to show G'' must be as chosen in the proof of Lemma 4.6.11.

If a pair of non-heavy vertices $u \in V'_i$ and $v \in V'_j$ are not connected in G, their cluster vertices in G'' cannot be connected, i.e. it must be the case that $\{i,j\} \notin E'$. If all pairs with disconnected clusters in G'' were disconnected, it would be uniquely determined, proving the claim. G is a monotone blowup, so as mentioned earlier, we must show that the adverse effect of heavy vertices and unnecessary edges cannot bring it into monotone agreement with a graph in Π'_m other than G''.

To do so, we show that the minimum number of (ordered) pairs of clusters, whose bipartite graph is not full, is high enough to practically determine G''. To obtain this minimum number, we bound the total number of disconnected (ordered) pairs of non-heavy vertices in different clusters: There are at most $(0.5 + 10^5\Delta)n^2$ ordered pairs in G connected by an edge, overall; and at most $2|V_{\text{heavy}}|n$ pairs are incident upon heavy vertices, so the total number of disconnected pairs in different clusters is at least $(0.5-10^5\Delta)n^2-2|V_{\text{heavy}}|n-\sum_i |V_i|^2 > \big(0.5-(2\cdot 10^7+10^5)\Delta\big)n^2-\sum_i (n/m)^2 > \big(0.5-201\cdot 10^5\Delta\big)n^2 - n^2/m$. As $m > \Delta^{-1}$ and $(201\cdot 10^5+1)\Delta < 0.001$, this is at least $(0.5-0.001)n^2$. There must therefore be at least $0.499m^2$ (ordered) pairs of different clusters with missing edges between non-heavy vertices. As the average degree of G'' is no lower than $(0.5-\Delta)m$, there are at most $m^2 - \big((0.5-\Delta)m^2 + 0.499m^2\big) < (0.001+\Delta)m^2 < 0.003\binom{|V''|}{2}$ additional (ordered) pairs of clusters which may fail to be present as edges in G''. Thus any two graphs of order m, with subgraphs which can serve as G'', are at a distance of no more than 0.003 from each other, corresponding to a choice of the potential additional missing edges. We recall that Π'_m graphs are 0.4-far from each other (condition 3 in Lemma 4.6.1 met by graphs in Π'). Combining these two facts we conclude that the order-m graph, of which G'' is a subgraph, is determined up to isomorphism; and as G' can be such a graph, the order-m graph is necessarily some relabeling of G'.

Now, since the distance between any two potential G'' graphs is less than 0.01, the labeling of G' is determined up to an isomorphism fixing over $0.9m$ of the vertices (as per condition 4 of Lemma 4.6.1). It remains to show that an isomorphism on G', which fixes over $0.9m$ of the vertices, but moves at least one vertex of a cluster i with $V_i \neq \varnothing$, makes G'' incompatible with $V'_1, \ldots, V'_m$.

Consider, therefore, two subgraphs G''_1 and G''_2 of relabelings G'_1 and G'_2 of G', satisfying the requirements with respect to the clustering $V'_1, \ldots, V'_m$, and such that the mapping between G'_1 and G'_2 fixes $0.9m$ of the vertices, but replaces a vertex $i \in V(G''_1)$ by some other vertex $j \neq i$ of G' (not necessarily a vertex in G'_2). The neighborhood of each non-heavy vertex in V'_i must contain all non-heavy vertices in clusters V'_k such that $\{i,k\} \in E'_1$, due to the monotone agreement with G''_1; and it must contain all vertices in clusters V'_k such that $\{j,k\} \in E'_2$, due to the monotone agreement with G''_2. Now, for at least a 0.9-fraction of the clusters V'_k, it is the case that $\{j,k\} \in E'_2$ if and only if $\{j,k\} \in E'_1$ — so the V'_i vertex' neighborhood must contain these non-heavy vertices.

Consequently, the degree of a vertex of V_i' must be at least $(0.5-\Delta)n$ (edges mandated by G_1''), plus $(0.5-\Delta-0.1)n$ (edges mandated by G_2'' to vertices fixed by the isomorphism), minus at most $(0.25+\Delta)n$ (the maximum intersection of the neighborhoods of pairs of vertices in G', by Lemma 4.6.1), minus $|V_{\text{heavy}}|n$ (heavy neighbors). Recalling that $|V_{\text{heavy}}|n \leq 10^7\Delta n$, this sum is at least $\big(0.65-\Delta-10^7\Delta\big)n > 0.64n$, which is impossible for a non-heavy vertex. Thus the isomorphism cannot replace any $i \in V(G_1'')$, so G_1'' and G_2'' must be exactly the same as labeled graphs, i.e. G'' is indeed uniquely determined, as claimed. □

We now have all the machinery necessary for presenting a test for Π^q, listed as Algorithm 4.4, and establishing its validity.

Algorithm 4.4 A test for Π^q

$\varepsilon' \leftarrow \min\{\varepsilon, \Delta/1000\}/20$, $m \leftarrow \lfloor\sqrt{q(n)}\rfloor$.

Phase I: Graph edge density estimation

Estimate the edge density of G, using $\Theta\big(1/\varepsilon'^2\big)$ independent edge queries.
If the estimated edge density exceeds $0.5+2\Delta-2\varepsilon'$, accept.

Phase II: Vertex degree estimation

$S_{\text{min-deg}} \leftarrow$ uniform sample of $\Theta(1/\varepsilon')$ vertices.
$S_{\text{sig}} \leftarrow$ uniform sample of $\Theta\big(\log(|S_{\text{min-deg}}|)/\varepsilon'^2\big)$ signature vertices.
for each vertex $v \in S_{\text{min-deg}}$ **do**
  Estimate the degree of v using S_{sig} (by querying the potential edges from v to S_{sig}).
  If v has estimated degree under $(0.5-\Delta-\varepsilon')n$, reject.
end for

Phase III: Finding representatives for a clustering

$S_{\text{rep}} \leftarrow$ uniform sample of $\Theta\big(m/\varepsilon'^2\big)$ vertices.
$S_{\text{sig}} \leftarrow$ uniform sample of $\Theta\big(\log(|S_{\text{rep}}|)/\varepsilon'^2\big)$ signature vertices.
$S'_{\text{rep}} \leftarrow \varnothing$
for each vertex $v \in S_{\text{rep}}$ **do**
  Estimate v's degree using S_{sig}.
  If v's estimated degree is less than $(0.52-\varepsilon')n$, add v to S'_{rep}.
end for
If $|S'_{\text{rep}}| < 0.99|S_{\text{rep}}|$, reject.
$S_{\text{sig}} \leftarrow$ uniform sample of $\Theta(\log(|S_{\text{rep}}|))$ signature vertices.
$m' \leftarrow 0$, $R \leftarrow \varnothing$
for each $v \in S'_{\text{rep}}$ **do**
  for each $i \in [m']$ **do**
    Estimate the size of the difference between the neighborhoods of v and r_i by the difference of their neighborhoods in S_{sig} .
    If the neighborhoods of v and r_i differ by no more than $0.06s$, add v to V_i'' and continue to the next iteration at line 20.
  end for
  $m' \leftarrow m+1$, $r_i \leftarrow v$, $V''_{m'} \leftarrow \{v\}$, $R \leftarrow R \cup \{r_i\}$
  If $m' > m$, reject.
end for
... (continued) ...

Observation 4.6.13. The queries made by Algorithm 4.4 are dominated by those in Phases IV and V: $\binom{m}{2} = \Theta(q(n))$ and $\Theta\big(t\cdot\log(t/\varepsilon)/\varepsilon^2\big) = \Theta\big(m\cdot\log^2(m)\cdot\varepsilon^{-4}\log(1/\varepsilon)\big)$ respectively. Thus the overall number of queries, ignoring the dependence on ε, is $\Theta(q(n))$.

Algorithm 4.4 A test for Π^q (continued)

Phase IV: Determining G' and G''

Query the graph G_R, induced by R and labeled accordingly.
Let $G' = ([n], E') \in \Pi'_m$ be such, that G_R is in monotone agreement with the subgraph induced by its first m' vertices: If $\{i, j\} \in E'$ then $\{r_i, r_j\} \in E$.
Let G'' be the subgraph of G' induced on the first m' vertices.
If there exists no appropriate G', or if G'' is not uniquely determined by G_R, reject.

Phase V: Estimating cluster sizes

$S_{\text{csize}} \leftarrow$ uniform sample of $t = \Theta\big(m\log(m) \cdot \log(1/\varepsilon')/\varepsilon'^2\big)$ vertices.
$S_{\text{sig}} \leftarrow$ uniform sample of $\Theta\big(\log(t/\varepsilon)/\varepsilon'^2\big)$ signature vertices.
for each $v \in S_{\text{csize}}$ **do**
  Estimate v's degree using S_{sig}.
  If v has estimated degree over $(0.52 - 2\varepsilon')n$, remove it from S_{csize} and continue to the next v.
  $\pi(v) \leftarrow \perp$
  for each $i \in [m']$ **do**
    Estimate the size of the difference between the neighborhoods of v and r_i using S_{sig}.
    If the neighborhoods of v and r_i differ by less than $0.06s$, let $\pi(v) = i$.
  end for
end for
If any cluster i has over $(1 + \varepsilon'/2)t/m$ vertices in S_{csize} with $\pi(v) = i$, reject.
If more than an $\varepsilon'/2$-fraction of the vertices remaining in S_{csize} have $\pi(v) = \perp$, reject.

Phase VI: Ensuring the monotone agreement of G with G''

$S_{\text{sig}} \leftarrow$ uniform sample of $\Theta\big(\log(1/\varepsilon')/\varepsilon'^2\big)$ signature vertices.
for $\Theta(1/\varepsilon')$ times **do**
  Sample a pair of vertices u, v and query $\{u, v\}$.
  Estimate the degrees of u and v using S_{sig}.
  If u or v have estimated degree over $(0.52 - \varepsilon')n$, continue to the next pair.
  Cluster u and v as in Phase V.
  If $\pi(u) = \perp$ or $\pi(v) = \perp$, continue to the next pair.
  If $\{\pi(u), \pi(v)\} \in E''$ but $\{u, v\} \notin E$, reject.
end for
accept.

In proving Algorithm 4.4's validity as a test, we will separate the arguments for completeness and for soundness, both of which being non-trivial.

Completeness of the test

We will again denote by V_{heavy} the set of heavy vertices of G; we also refer to vertices with degree under $0.52 - 2\varepsilon'$ as *light* vertices; the rest are *non-light* vertices, and the set of these vertices is denoted V_{nl}.

Lemma 4.6.14. *A graph of order n which satisfies (C2) but not (C1) has at most $150\Delta n$ vertices of degree over $(0.52 - 3\varepsilon')n$. In particular, less than a 150Δ-fraction of its vertices are non-light, i.e. $|V_{nl}| < 150\Delta n$.*

Proof. The argument is similar to that made for heavy vertices earlier in this section: Let U denote the set of vertices with degree over $0.52 - 3\varepsilon'$. Since the average degree of G is less than $(0.5 + 2\Delta)$, and the minimum degree is at least $(0.5 - \Delta)n$, U satisfies $(0.52 - 3\varepsilon')n \cdot |U| + (0.5 - \Delta)n \cdot (n - |U|) \leq (0.5 + 2\Delta)n^2$; thus $(0.02 + \Delta - 3\varepsilon')n \cdot |U| \leq (3\Delta)n^2$; this yields the claim, as $0.02 + \Delta - 3\varepsilon' > 0.02$. □

Lemma 4.6.15. *If $G \in \Pi^q$, then it is accepted by Algorithm 4.4 with probability at least* 2/3.

Proof. If G satisfies condition (C1), then it will be accepted with high probabilityby Phase I of the test; in fact, this will be true if G's average degree is at least $(0.5 + 2\Delta - \varepsilon')n$. We thus focus on the case of G satisfying Π^q but having less than this average degree, thus satisfying (C2). Let $G' = ([m], E') \in \Pi^q$ be the graph of which G is an approximate monotone blowup. To prove that the test accepts G with high enough probability, we show that each of the following 'desirable' events is likely to occur:

1. The graph is not determined to have low minimum degree (in Phase II).
2. Almost all vertices sampled into S_{rep} are light vertices, which are then placed in S'_{rep} (hence Phase III does not reject on account of S'_{rep} being too small).
3. The clustering of S'_{rep} in Phase III is valid, i.e. the vertices assigned to each cluster are all those vertices of S'_{rep} in the cluster of some single G' vertex.
4. By the end of Phase III, R contains only non-heavy vertices, and its light vertices represent almost all clusters of G'.
5. The graph G_R, induced by the cluster representatives in R, is such that its corresponding G'' is uniquely determined (hence Phase IV does not reject).
6. All cluster size estimates in Phase V are about $1/m$ of the total size of S_{csize} (hence Phase V does not reject on account of cluster size imbalance),
7. The clustering of S_{csize} is valid, and all heavy vertices are discarded (hence Phase V does not reject on account of there being too many unclusterable vertices).
8. The fraction of S_{csize} discarded for having high degree is not excessively high.
9. The clustering of pairs in Phase VI is valid, i.e. all pairs with a heavy endpoint are discarded, and all vertices in pairs assigned $\pi(v) = i$ are non-heavy vertices from the same G' vertex cluster as the representative r_i.
10. Phase VI finds no monotone disagreement between G and G'' (and hence does not reject).

If all of these events occur, G is indeed accepted.

Phase II degree estimates The degree estimate of a single vertex $v \in S_{\text{min-deg}}$ is ε'-close to its actual value with probability $1 - \exp\bigl(-\Omega\bigl(\varepsilon'^2 \cdot |S_{\text{sig}}|\bigr)\bigr) = 1 - \exp\bigl(-\Omega(\varepsilon'^2 \cdot \varepsilon'^{-2} \cdot \log(|S_{\text{min-deg}}|))\bigr) > 1 - 0.01 \cdot \bigl|S_{\text{min-deg}}\bigr|$. (This last argument uses a large-deviation bound on the vertices in in $S_{\text{min-deg}}$, which are uniform samples without repetition; see the note on page 68, following the proof of Lemma 4.5.17.) Union-bounding over all vertices in $S_{\text{min-deg}}$ we conclude that with probability greater than $1 - 0.01$, all of their degree estimates are correct to within ε'. As G's minimum degree is at least $(0.5 - \Delta)n$, the estimates are all at least $(0.5 - \Delta - \varepsilon')n$, so G is not rejected at Phase II in this event.

Light vertices in S_{rep} and S'_{rep} In Phase III, the degree estimate of a single vertex of S_{rep} is ε'-close to its actual value with probability $1 - \exp(-\Omega(\log(|S_{\text{rep}}|)))$ (independently of the choice of S'_{rep}), so with probability greater than $1 - 0.005$ all s vertices in the set have estimates correct to within ε'. Also, with probability greater than $1 - \exp(\Omega(m)) > 1 - 0.005$, the fraction of non-light vertices in S_{rep} is at most ε' higher than their fraction in G (which is under 150Δ, by Lemma 4.6.14); in this event, $|S_{\text{rep}}|$ has at most a $150\Delta + \varepsilon' < 151\Delta < 0.01$ fraction of non-light vertices. If both events occur, more than $0.99|S_{\text{rep}}|$ light vertices are placed in S'_{rep}, so the test does not reject on account of S'_{rep} being too small. Note also that these events occur with probability greater than $1 - 0.01$, independently of the choice of light vertices in S_{rep} given their total number; in other words, with probability greater than $1 - 0.01$ these events occur and, additionally, if we condition on the specific number s of light vertices in S_{rep}, these light vertices are distributed uniformly over all sets of s light vertices in S_{rep}.

Validity of the clustering of S'_{rep} Let $V'_1, \ldots, V'_m$ be as in Lemma 4.6.11. Now, two non-heavy vertices in S'_{rep} in different V'_i's have neighborhoods differing on $0.45n$ vertices, by item 7 in Lemma 4.6.11, and two non-heavy vertices in the same V'_i have neighborhoods differing on at most $0.05n$ vertices by item 6 in Lemma 4.6.11. Thus with probability $1 - \exp(\Omega(\log(|S_{\text{rep}}|)))$, a pair of non-heavy vertices will be estimated to have neighborhoods with under $0.06n$ differences, if the pair of vertices are in the same cluster, and over 0.06 if they are in different clusters, using the set of s' signature vertices. Thus with probability greater than $1 - 0.005$, all decisions of whether the pairs of non-heavy vertices in S'_{rep} are in the same cluster will be correct — independently of which non-heavy vertices make up S'_{rep}. As argued above, with probability greater than $1 - 0.005$ all degree estimates of S_{rep} are correct, so no heavy vertices are placed in S'_{rep}. Thus with probability greater than $1 - 0.01$ all clustering decisions regarding pairs of vertices in S'_{rep} are correct, the clustering is valid, and the test will not reject G on account of having more than m clusters. This, independently of the choice of light vertices in S_{rep} given their total number (see comment above).

R represents most clusters well Suppose that all degree estimates in Phase III are correct to within ε'. In this case no heavy vertex is placed in S'_{rep} and consequently no heavy vertex is placed in R. It then remains to show that few clusters are missing representatives in R.

Now, suppose additionally that the clustering of S'_{rep} is valid. In this case, if a cluster is represented in S'_{rep}, it will be represented in R and not have its constituent non-heavy vertices represented by some r_i from another cluster. Finally, suppose that light vertices in S_{rep} are all placed in S'_{rep}. With this supposition it suffices to show that only few clusters have no light vertices in S_{rep}.

Now, there are less than $150\Delta n$ non-light vertices in G, and thus less than $150\Delta/(1-$

$\varepsilon') < 151\Delta - \varepsilon'$ clusters have fewer than $\varepsilon' n/16m$ light vertices (we refer to such clusters as *light-poor*). A cluster with at least $\varepsilon' n/16m$ light vertices (a *light-rich* cluster) has at least one light vertex in S_{rep} with probability at least $1 - \exp(-\Omega(1/\varepsilon'))$, thus the expected number of light-rich clusters, having no light vertices in S_{rep}, is less than $\varepsilon' m/3200$. If this is the case, then with probability greater than $1 - 0.005$, the actual number of such light-rich clusters is less than $\varepsilon' m/16$, making the total number of clusters missing from R lower than $151\Delta m$, i.e. $m - m' < 151\Delta m$. When the above holds, and R has no heavy vertices, we refer to R as being *well-representing*.

Thus R is well-representing with probability greater than $1 - 0.03$ overall, and with probability greater than $1 - 0.005$ conditioning on relevant previous desirable events. This event occurs with probability greater than $1 - 0.03$, independently of the distribution of light vertices within each cluster given their number in that cluster. (Note, however, that the number of light vertices in each cluster is not independent of the event of R being well-representing).

The uniqueness of G'' We would like to show that the graph G_R will be found to uniquely determine G'', the subgraph of G' corresponding to the clusters represented in R. We condition on the event of R being well-represented, with the choice of light vertices in S_{rep} being uniform given their total number. When this event occurs, the appropriate subgraph of G' is necessarily a possible choice for G'' at Phase IV, as it is in monotone agreement with G_R — but we wish to show that it is the only such choice, using Lemma 4.6.12.

We cannot apply Lemma 4.6.12 to G itself, which is unknown to the test, nor to G_R. Instead, consider a graph $\widetilde{G}$ obtained as follows: We blow up G_R by a factor of n/m; for any unrepresented cluster in G, we add a cluster of n/m heavy vertices, with full degree $n - 1$. We will demonstrate that $\widetilde{G}$ is in Ξ, so that Lemma 4.6.11 and Lemma 4.6.12 apply to it. This will establish the uniqueness of our desired G'', as $\widetilde{G}$ is its approximate monotone blowup.

Regarding the minimum degree of $\widetilde{G}$: Had R represented all clusters, the minimum degree would have been $(0.5 - \Delta)n$, as $\widetilde{G}$ would have been a monotone blowup of G' with no heavy vertices. Since we replace missing clusters with full-degree vertices, edges are only added relative to the case of having more clusters represented. Thus the minimum degree is no less than $(0.5 - \Delta)n$ in $\widetilde{G}$ as well.

Now let us bound the average degree of $\widetilde{G}$, letting d_H denote the average degree of a graph H. As $\widetilde{G}$ is a random graph based on the choice of R, let us consider the distribution of a single representative r_i. The representative is not a uniform sample from V_i, as a uniform sample may fail to be estimated as non-heavy even with our having conditioned on R being well-representing. But with our conditioning, if r_i is a *light* vertex, its distribution is uniform over all light vertices in the cluster. Thus the variation distance between the distribution of any r_i, and the uniform distribution over its cluster V_i, is at most the probability of the uniformly-sampled V_i vertex being

non-light. This implies, specifically, that for two represented clusters i and j, with r_i and r_j being their (random-variable) representatives

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{r_i,r_j}[\{r_i,r_j\}\in E]\\
&\leq \mathop{\mathbf{Pr}}_{\substack{(v_i,v_j)\in V_i\times V_j\\ \text{uniform}}}[\{v_i,v_j\}\in E] + \mathop{\mathbf{Pr}}_{\substack{(v_i,v_j)\in V_i\times V_j\\ \text{uniform}}}[v_i \text{ or } v_j \text{ non-light}]\\
&\leq \mathop{\mathbf{Pr}}_{\substack{(v_i,v_j)\in V_i\times V_j\\ \text{uniform}}}[\{v_i,v_j\}\in E] + \mathop{\mathbf{Pr}}_{\substack{v_i\in V_i\\ \text{uniform}}}[v_i \text{ non-light}] + \mathop{\mathbf{Pr}}_{\substack{v_j\in V_j\\ \text{uniform}}}[v_j \text{ non-light}]
\end{aligned}$$

Also, had all clusters been represented in R, and had r_i been sampled uniformly from its cluster, the expected average degree in G_R would be exactly the average degree of G (normalized by m/n), i.e. is at most $(0.5+2\Delta-\varepsilon')m$.

Bearing the above in mind, we can obtain a bound on the number of tuples in G_R (which is m' times its average degree) if we account for non-light vertices. Recall that the expectation is under our conditioning of R to be well-representing.

$$\begin{aligned}
&\mathbf{Ex}\big[m'\cdot d_{G_R}\big]\\
&= \sum_{i,j\in[m]} \mathbf{Pr}[i,j \text{ represented}]\cdot \mathop{\mathbf{Pr}}_{r_i,r_j}[\{r_i,r_j\}\in E | i,j \text{ represented}]\\
&\leq \sum_{i,j\in[m]} \left(\mathop{\mathbf{Pr}}_{\substack{(v_i,v_j)\in V_i\times V_j\\ \text{uniform}}}[\{v_i,v_j\}\in E] + \mathop{\mathbf{Pr}}_{\substack{v_i\in V_i\\ \text{uniform}}}[v_i \text{ non-light}] + \mathop{\mathbf{Pr}}_{\substack{v_j\in V_j\\ \text{uniform}}}[v_j \text{ non-light}] \right)\\
&= \sum_{i,j\in[m]} \mathop{\mathbf{Pr}}_{\substack{(v_i,v_j)\in V_i\times V_j\\ \text{uniform}}}[\{v_i,v_j\}\in E] + 2m\cdot \sum_{i\in[m]} \mathop{\mathbf{Pr}}_{\substack{v_i\in V_i\\ \text{uniform}}}[v_i \text{ non-light}]\\
&\leq \frac{m^2}{n} d_G + 2\cdot\left(\frac{m^2}{n}\cdot|V_{\text{nl}}|\right) \leq m\cdot\left(\frac{m}{n}d_G + 300\Delta m\right)
\end{aligned}$$

The expected degree of $\widetilde{G}$ is n/m times that of G_R, plus less than n/m for every one of the $m-m'$ clusters unrepresented in R. As R is well-representing, $m-m' < 151\Delta m$, so the contribution of unrepresented clusters to the expected degree is at most $(n/m)\cdot(151\Delta m) = 151\Delta n$. Thus,

$$\begin{aligned}
\mathbf{Ex}\big[d_{\widetilde{G}}\big] &< \mathbf{Ex}\Big[\frac{n}{m}\cdot d_{G_R} + 151\Delta n\Big] = \mathbf{Ex}\Big[\frac{n}{m\cdot m'}\cdot\big(m'\cdot d_{G_R}\big) + 151\Delta n\Big]\\
&< \frac{n}{(1-151\Delta)m^2}\cdot\mathbf{Ex}\big[\big(m'\cdot d_{G_R}\big)\big] + 151\Delta n\\
&< (1+302\Delta)\frac{n}{m^2}\cdot m\cdot\Big(\frac{m}{n}d_G + 300\Delta m\Big) + 151\Delta n\\
&< (1+302\Delta)\cdot(d_G + 300\Delta n) + 151\Delta n\\
&< (1+302\Delta)\cdot\big((0.5+2\Delta-\varepsilon')n + 300\Delta n\big) + 151\Delta n\\
&< (0.5 + (2 + 302\cdot 0.5 + 302\cdot 2\Delta + 300 + 302\cdot 300\Delta + 151)\Delta)n\\
&< (0.5+605\Delta)n
\end{aligned}$$

As the average degree of $\widetilde{G}$ cannot be lower than its minimum degree, which is at least $(0.5-\Delta)n$, we can apply Markov's inequality to the difference between the average and the minimum degrees of G_R to conclude that, with probability greater than $1-0.2$, the average degree of $\widetilde{G}$ is under $0.5+5000\Delta$. When this occurs, Lemma 4.6.11 and Lemma 4.6.12 apply to $\widetilde{G}$.

Thus an appropriate G' and G'' exist, with G'' determined uniquely, and the test doesn't reject in Phase IV — with probability greater than $1-0.23$ overall, and with probability greater than $1-0.2$ conditioned on those relevant previous desirable events occurring.

Validity of the clustering of S_{csize} We can employ the same argument as for the clustering S_{rep}, except that the set size is t: With probability greater than $1-0.005$ all degree estimates are correct to within ε', and with probability greater than $1-0.005$, all decisions of whether the pairs of non-heavy vertices in S'_{rep} and in S'_{csize} are in the same cluster will be correct. Thus supposing that the representatives in R are all non-heavy, with probability greater than $1-0.01$ all heavy vertices are discarded and all clustering decisions are correct — that is, for every vertex v remaining in S_{csize}, $\pi(v)=i$ if v is an in the cluster of r_i, and $\pi(v)=\perp$ if it is unrepresented.

Thus the clustering is valid with probability greater than $1-0.04$ overall, and with probability greater than $1-0.01$ conditioning on relevant previous desirable events.

Cluster sizes in S_{csize} As every cluster in G corresponding to a vertex of G' has size n/m, the expected fraction of S_{csize} from each cluster is $1/m$. We apply a large deviation bound for sums of low-probability indicators (see, e.g. [ASE92, Theorem A.11]) to conclude that the probability of a cluster having more than $(1+\varepsilon'/2)t/m$ vertices is less then $\exp\big(-(\varepsilon'\mathrm{t}/2\mathrm{m})^2/2(\mathrm{t}/\mathrm{m})+(\varepsilon'\mathrm{t}/2\mathrm{m})^3/2(\mathrm{t}/\mathrm{m})^2\big) = \exp\big((-\varepsilon'^2+\varepsilon'^3/2)\cdot \mathrm{t}/8\mathrm{m}\big) < \exp\big(-\varepsilon'^2\cdot \mathrm{t}/10\mathrm{m}\big) = \exp(-\Omega(\log(\mathrm{m}))) < 0.01/\mathrm{m}$. Union-bounding over all m clusters we find that with probability greater than $1-0.01$, all clusters have less than $(1+\varepsilon'/2)t/m$ vertices in S_{csize}. When this event occurs and the clustering is also valid, no cluster has more than $(1+\varepsilon'/2)t/m$ vertices assigned the same $\pi(\cdot)$ value.

Thus the cluster sizes are all determined to be within $\varepsilon'/2$ of their expected value, and the test does not reject on account of cluster size imbalance, with probability greater than $1-(0.04+0.01)=1-0.05$ overall, and with probability greater than $1-0.01$ conditioning on relevant previous desirable events.

High-degree vertices in S_{csize} Vertices in S_{csize} may only be discarded if they are estimated to have degree over $0.52-2\varepsilon'$. With probability greater than $1-0.005$, all degree estimates of vertices in S_{csize} are correct to within ε' (by an argument similar to previous phases) — independently of the choice of vertices in S_{csize}. When this event occurs, only vertices with degree over $0.52-3\varepsilon'$ may be discarded. By Lemma 4.6.14, the fraction of these is at most 150Δ. By a similar argument as for Phase III, the fraction

of these vertices in S_{csize} is at most ε' higher than their fraction in G, with probability greater than $1 - 0.005$. Thus with probability greater than $1 - 0.01$ (regardless of previous desirable events), less than a $150\Delta + \varepsilon' < 151\Delta$-fraction of the vertices of S_{csize} are discarded for having overly high degree, independently of the choice of these vertices in S_{csize}.

Note that this event is not one of the desirable events listed earlier in the proof, but it is useful to condition upon; see below.

Vertex clustering failures in S_{csize} We wish to bound the fraction of vertices in S_{csize} which are not discarded for overly high degree, but are assigned $\pi(v) = \perp$; we suppose that R is well-representing.

To do so, we begin by bounding the fraction of light vertices in S_{csize} without a representative in R. There are at most $\varepsilon' m/16$ light-rich clusters unrepresented in R; thus the total number of light vertices whose clusters are unrepresented in R does not exceed $m \cdot \varepsilon' n/16m$ in light-poor clusters, plus $\varepsilon' m/16 \cdot n/m$ in light-rich clusters, or $\varepsilon' n/8$ overall. Consequently, the expected fraction of light vertices in S_{csize} unrepresented in R is $\varepsilon'/8$; with probability greater than $1 - \Omega(m \cdot \log(1/\varepsilon)) > 1 - 0.005$, the actual fraction is under $\varepsilon'/4$.

Suppose that the vertex degree estimates in Phase V are correct to within ε' (this happens with probability greater than $1 - 0.005$ independently of the choice of S_{csize}). Suppose also that the fraction of vertices with degree over $0.52 - 3\varepsilon'$ in S_{csize} (before any discards) is at most 151Δ; as argued above, this occurs with probability greater than $1 - 0.005$. When both these events occur, the number of light vertices discarded for having overly high estimated degree is at most a 151Δ-fraction of the vertices of S_{csize}. Thus the fraction of unrepresented light vertices after the discard is at most $1/(1 - 151\Delta) < 2$ times the original fraction, i.e. under $\varepsilon'/2$. As all vertices remaining after the discard are light, the fraction of unrepresented light vertices is the fraction of unrepresented vertices remaining in S_{csize}. Finally, if we suppose that the clustering in Phase V is valid, this fraction is the fraction of vertices v remaining in S_{csize} with $\pi(v) = \perp$.

Thus less then an $\varepsilon'/2$-fraction of the vertices remaining in S_{csize} are assigned $\pi(v) = \perp$, with probability greater than $1 - 0.06$ altogether, and with probability greater than $1 - 0.01$ conditioning on relevant previous desirable events.

Validity of the clustering in Phase VI The argument regarding the clustering in Phase III applies also to the clustering of vertices in Phase VI, using the representatives in R. If the representatives in R are all non-heavy, then with probability greater than $1 - 0.005$, every one of the non-heavy vertices being clustered in Phase VI will be assigned the correct cluster, or assigned $\perp$ if their cluster is not represented in R. With probability greater than $1 - 0.005$ all vertices sampled in Phase VI have degree estimates correct to within ε', so no clustering is attempted of heavy vertices. Thus the clustering

in Phase VI is valid with probability greater than $1-(0.03+0.01)=1-0.04$ overall, and with probability greater than $1-0.01$ conditioned on relevant previous desirable events occurring.

Monotone agreement of G with G'' Suppose that G'' is determined uniquely, and that the clustering in Phase VI is valid. When both these events occur, Phase VI does not reject, because non-heavy vertices in G are in monotone agreement with G', and the test only checks vertices (correctly) determined to be non-heavy.

Thus G is not rejected in Phase VI with probability greater than $1-0.24$ overall, and with probability 1 conditioning on relevant previous desirable events.

The conjunction of all desirable events above occurs with probability greater than $1-0.305>2/3$, so the test indeed accepts with sufficient probability. □

Note. As in Section 4.5, the large deviation bounds are applied as though the vertex samples are independent, while when a set of vertices is sampled without repetitions, this is not the case. However, such bounds are even tighter for samples without repetitions, so such use is justified.

Soundness of the test

Lemma 4.6.16. *If G is ε-far from Π^q, then it is rejected by Algorithm 4.4 with probability at least $2/3$.*

Proof. We prove that if the test accepts with probability at least $1/3$, then G cannot be ε-far from Π^q.

If G has average degree over $(0.5+2\Delta-\varepsilon)n$, then it isn't ε-far from Π^q and the claim holds trivially. We thus assume G's average degree is under $(0.5+2\Delta-\varepsilon)n$. In this case, G is accepted with at most a small constant probability in Phase I; it thus remains to prove that if the other phases accept with probability at most slightly lower than $1/3$, then G cannot be ε-far from Π^q.

If G has more than $\varepsilon' n$ vertices with degree under $(0.5-\Delta-2\varepsilon')n$, then it is rejected by Phase II, the vertex degree estimation phase, with probability greater than $3/4$, and the claim holds. Let us also assume, therefore, that G has at most $\varepsilon' n$ vertices with degree under $(0.5-\Delta-2\varepsilon')n$.

Let S'_{rep}, R and G'' be as determined in Phases III and IV. The clustering π they induce is a clustering of at least $(1-\varepsilon')n$ of the vertices with degree at most $0.52-3\varepsilon'$ (that is, at most ε' of these have $\pi(v)=\perp$), as otherwise Phase V rejects with high probability. Also, each cluster contains at most $(1+2\varepsilon')\cdot n/m$ such vertices, as otherwise Phase VI rejects with high probability. Finally, the number of edges missing between clusters, which are connected in G'', is at most $\varepsilon'\binom{n}{2}$, as otherwise, again, Phase V rejects with high probability.

To complete the proof we show that under the above conditions, the graph is close to satisfying either (C1) or (C2), and consequently close to Π^q. Indeed, suppose that we modify G as follows:

1. Add the edges missing between clusters connected in G'' edges.
2. Move vertices with degree at most $0.52 - 3\varepsilon'$ from clusters with more than n/m vertices to smaller clusters (at most $2\varepsilon' n$ such vertices need be moved), adding to their neighborhoods those edges mandated by G' for the new cluster.
3. Move vertices with degree at most $0.52 - 3\varepsilon'$, which have $\pi(v) = \perp$, into any cluster into clusters with less than n/m vertices (including possibly clusters not represented in R), adding to their neighborhoods those edges mandated by G' for the new cluster.
4. Arbitrarily add edges to vertices with degree at least $0.52 - 3\varepsilon'$ to make them heavy (i.e. increase their degree to 0.52).
5. Distribute heavy vertices among clusters so that each cluster has exactly n/m vertices (without making any actual edge changes).

The result of the above modifications is a partition into m equal-size cells, which constitutes an approximate monotone blowup of G'. Finally, we connect heavy vertices to low-degree vertices until they meet the minimum degree requirement in (C2). This is possible, due to the fact that in a monotone blowup of G' to order n with no heavy vertices, each vertex has degree at least $(0.5 - \Delta)n$, so any vertex in the modified G with degree lower than $(0.5 - \Delta)n$ must be missing edges necessitated by G'; these cannot be missing between pairs of non-heavy vertices, due to the monotone agreement of G with G', so they must be missing between heavy and non-heavy vertices.

Now, if at any point in the above operations we have added so many edges, that the average degree of the modified G surpasses $0.5+2\Delta$ — the graph satisfies Π^q by condition (C1), and we leave it as it is; otherwise, after all these operations, the graph must satisfy (C2). Either way, the number of edges we have added is at most: $\varepsilon'\binom{n}{2}$ for the first operation; $(2\varepsilon' n) \cdot (n-1) = 4\varepsilon'\binom{n}{2}$ for the second operation; $(\varepsilon' n) \cdot (n-1) = 2\varepsilon'\binom{n}{2}$ for the third operation; $3\varepsilon' n^2 < 7\varepsilon'\binom{n}{2}$ for the fourth operation (for sufficiently large n); no changes for the fifth operation on the list; and less than $(2 + (0.5 + \Delta))\varepsilon' n^2 < 6\varepsilon'\binom{n}{2}$ for the final minimum degree increase (for sufficiently large n). In total, less than $(1+4+2+7+6)\varepsilon'\binom{n}{2} = 20\varepsilon'\binom{n}{2} = \varepsilon\binom{n}{2}$ edge additions are necessary to make G satisfy either (C1) or (C2). G is therefore not ε-far from Π^q. □

Observation 4.6.17. The test in Algorithm 4.4 can be generalized to the case in which n/m is not an integer. The modification required, in broad terms, is to check that about $n \bmod m$ of the vertices have full-degree, account for them as an $(m+1)^{\text{th}}$ cluster, and set the other cluster sizes accordingly. This may also necessitate special handling of heavy vertices with full or almost-full degree (which cannot be told apart from the 'real' $n \bmod m$ full-degree vertices) apart from heavy vertices with significantly lower degree. We do not go into the details.

Theorem 4.6 is now proven by combining the lower bound of Lemma 4.6.9, and the upper bound established through the test in Algorithm 4.4 — valid as per Lemma 4.6.15 and Lemma 4.6.16 — if m divides n, and by their variations taking Observation 4.6.17 into account, otherwise.

4.7 A hierarchy of one-sided-testable properties

We continue Section 4.5 and Section 4.6 with a third hierarchy theorem for dense graph properties. In this section, we modify the construction in Section 4.5, so as to make the properties amenable to a *one-sided* test at an arbitrary query complexity, while any significant reduction in the number of queries precludes even two-sided testing — in a sense, a tighter hierarchy. Unfortunately, while the construction maintains the **PTIME**-decidability of the property itself, it seems to make testing the property less computationally efficient, that is, we are not able to present a test whose running time is polynomial in its number of queries — as a test seems to need to decide what is essentially a subgraph isomorphism problem.

Theorem 4.7. *There exists a constant $\varepsilon_{4.6} > 0$, such that for every reasonable $q(\cdot)$ (in the sense of Definition 4.5.1), there exists a property of dense graphs that is testable with one-sided error using $O\big(q(n)/\varepsilon^2\big)$ queries (or $O(q(n))$ queries ignoring ε), but not ε-testable using $o(q(n))$ queries, even allowing two-sided error, for $\varepsilon \leq \varepsilon_{4.7}$. Furthermore, if $q(n)$ is computable from n in $\mathrm{poly}(n)$ time, then the property is* **PTIME***-decidable.*

4.7.1 Property construction

Thinking about how to obtain a one-sided-testing hierarchy theorem, we naturally ask ourselves whether Algorithm 4.3, the test used for the upper bound in Section 4.5, can be made one-sided. The reason it cannot is that we require the clusters in the blown-up graphs there to be of equal or almost-equal size; and if the cluster sizes are off, with some clusters being significantly larger than others, then the graph would be far from an appropriate blow-up (since vertices cannot be moved from one cluster to another without many edge changes). A test cannot avoid, therefore, having to estimate these sizes — and this estimate can be invalid, as the test's sampled vertices may come from just a few of the clusters. In light of this fact, let us forego the strict requirement on cluster sizes, and only require that a graph be a *generalized* blowup (see Definition 2.3.6), with potentially highly-disparate cluster sizes, but keeping all clusters present for easy deterministic decision. (This modification will also allow us to handle more cleanly handle the possibility of n not being an integer multiple of the size of the pre-blown-up graph). We can then make a similar argument to that in Section 4.5, with the necessary allowance for this generalization:

Lemma 4.7.1. *There exists a global constant $c_{4.7.1} > 0$, such that for every n, ε, α and every pair of (unlabeled) graphs (G_1, G_2) of order n, with G_1 being α-dispersed, the following holds: If G_1 and G_2 are ε-far from each other, then any (relaxed or proper) generalized blowup of G_2 to order n is at least $c_{4.7.1}\alpha^2\varepsilon$-far from any balanced blowup of G_1.*

Before proceeding to the proof, we recall having established in Lemma 4.5.13 that a *balanced* blowup (rather than merely a relaxed generalized blowup) of G_2 must be $c_{4.5.13} \cdot \alpha \cdot \varepsilon$-far from a balanced blowup of G_1. To prove that a relaxed generalized blowup G_2' of G_2 is also far from G_1', we will want to relate the "degree of imbalance" of a blowup to its distance from *any* balanced blowup. To do so, we first formalize this concept.

Definition 4.7.2. Let G be a labeled graph of order n and G' a relaxed generalized blowup of G to order n'. The *(relative) weight* ρ_i of the cluster V_i of the i^{th} vertex of G is the fraction $|V_i|/n'$.

Definition 4.7.3. Let G be a graph of order n and G' a relaxed generalized blowup of G to order n', with $t = \lfloor n'/n \rfloor$. G' is said to be a δ-*balanced* blowup of G if the variation distance between the relative weights of clusters in the blowup, and the relative weights of clusters in a balanced blowup, is at most δ — over all possible choices of $n' \bmod n$ larger clusters in a balanced blowup, i.e.

$$\min\left\{ \sum_{i=1}^{n} \left| \rho_i - \frac{s_i}{n'} \right| \;\middle|\; s_i \in \{t, t+1\} \wedge \sum_{i=1}^{n} s_i = n' \right\} \leq 2\delta$$

Notes.
– A balanced blowup is 0-balanced, and any relaxed generalized blowup is 1-balanced.
– If n divides n', the condition for δ-balance is merely $\sum\left|\rho_i - \frac{t}{n'}\right| = \sum\left|\rho_i - \frac{1}{n}\right| \leq 2\delta$.

Proof of Lemma 4.7.1. Let G_1' be a balanced blowup of G_1 (with clusters of size either t or $t+1$) and G_2' be a relaxed generalized blowup of G_2'. Let us label the vertices of both graphs, so that we may denote $V(G_1) = V(G_2) = [n]$ (this also induces a labeling of the blowup clusters)

We distinguish two cases, based on the "degree of imbalance" in the blowup of G_2 into G_2'. Our threshold δ-balance value for the analysis will be $\delta = c_{4.5.13} \cdot \alpha \cdot \varepsilon/5$.

Suppose, first, that G_2' is a δ-balanced blowup of G_2. If that is the case, G_2' is in fact 4δ-close to a balanced blowup of G_2: For the choice of s_i achieving the variation distance, one simply moves a 2δ fraction of the vertices between clusters of G_2', so that the cluster sizes become exactly the chosen s_i values. Switching the cluster of a single vertex entails as many as $n'-1$ edge changes, for a total of $2\delta n'(n'-1) = 4\delta\binom{n'}{2}$ over all vertices moved. We now use the triangle inequality to conclude that the δ-balanced blowup is at least $(c_{4.5.13} \cdot \alpha \cdot \varepsilon - 4\delta)$-far from G_1'; and we note that $c_{4.5.13} \cdot \alpha \cdot \varepsilon - 4\delta = \delta = c_{4.5.13} \cdot \alpha \cdot \varepsilon/5$.

We are therefore left with the case of G'_2 not being δ-balanced. If, for every bijection between G'_1 and G'_2, we were somehow able to pair the 'excess' vertices with other distinct vertices, so that every pair is in the same cluster of G'_2 but with different origins in G'_1, our proof would be concluded, as any such pair entails many discrepancies with respect to the bijection.

Towards this end, note first that due to the δ-imbalance of G'_2, for every choice of blowup cluster sizes, more than a δ-fraction of the weight of clusters is excess weight beyond the designated cluster weight, i.e. for every choice of $(s_i)_{i=1}^n$ corresponding to a blowup, and denoting $I = \left\{i \in [n] \;\middle|\; \rho_i > \frac{s_i}{n'}\right\}$, we have $\sum_{i\in I}\left(\rho_i - s_i/n'\right) > \delta$.

Now consider some bijection π' between the two blowups. Let $s_i^{\pi'} = t+1$ if any G'_1 clusters of size $t+1$ are mapped to i in their entirety and t otherwise. Clearly, there are at most $n' \bmod n$ indices i such that $s_i^{\pi'} = t+1$, so there exists some choice of s_i's corresponding to a blowup for which $s_i \geq s_i^{\pi'}$ for every i. Now, since for this choice we have $\sum_{i\in I}\left(\rho_i - s_i/n'\right) > \delta$, we also have, for the same I, $\sum_{i\in I}\left(\rho_i - s_i^{\pi'}/n'\right) > \delta$.

We now wish to 'pair up' vertices from different G'_1 clusters within clusters of G'_2. Consider some cluster i of G'_2, with $\rho_i n'$ vertices. The largest set of vertices in this cluster with the same origin in G'_1 is of size at most $s_i^{\pi'}$; consequently, cluster i has at least $\frac{1}{2}\left(\rho_i - s_i^{\pi'}/n'\right)$ pairs of vertices from different clusters. (To see why this is the case, think about repeatedly removing arbitrary pairs of vertices in G'_2 originating in different clusters of G'_1; eventually one is left with vertices in G'_2 all from the same cluster in G'_1, and their number cannot exceed $s_i^{\pi'}$.) Over all clusters in I, we have $\sum_{i\in I}\frac{1}{2}\left(\rho_i n' - s_i^{\pi'}\right) > \delta n'/2$ such pairs. Each pair is the cause of $\alpha t n > \alpha \cdot n'/2$ distinct discrepancies (as discussed in the proof of Lemma 4.5.13 — the neighborhoods of the two vertices must be made the same); the total number of discrepancies under π' due to all these pairs is at least $\delta/2 \cdot \alpha {n'}^2/2$. π' was chosen arbitrarily, so the same minimum number of discrepancies exists under any bijection between G'_1 and G'_2; thus the distance between the two graphs is at least $\delta\alpha/4$.

The claim is now proven by setting $c_{4.7.1} = c_{4.5.13}/20$ and noting that $\min(\delta, \delta\alpha/4) = \delta\alpha/4 = c_{4.5.13}\alpha^2/20$. □

A complexity-q property. Let Π'' be as constructed in Subsection 4.5.1, a dispersed **PTIME**-decidable property requiring $\Omega\left(n^2\right)$ queries, and let $m(n,q)$ be as in Definition 4.5.9. We set $\Pi^q = \bigcup_{n\in\mathbb{N}} {\Pi^q}_n$, with ${\Pi^q}_n$ containing all (proper) generalized blowups of graphs in ${\Pi''}_{m(n,q)}$. In other words, a graph in ${\Pi^q}_n$ has m non-empty clusters with complete bipartite graphs between cluster pairs corresponding to pre-blowup edges.

Lemma 4.7.4. *If $q(n)$ is computable from n in $\mathrm{poly}(n)$ time, then Π^q is* **PTIME***-decidable.*

Proof. The proof is very similar to that of Lemma 4.5.11: Since no two vertices of a graph in Π'' have the same neighborhood, one can easily reconstruct the original graph given m non-empty clusters, regardless of their sizes (but assuming that $q(n)$ itself can

be computed in polynomial time). Since Π'' is in **PTIME**, one can then efficiently decide whether the pre-blown-up graph satisfies it or not. Note that the fact that Π^q contains generalized blowups rather than *relaxed* generalized blowups is critical to this argument, as without a vertex from every cluster, one would only be able to reconstruct a subgraph of the original order-m graph, and might then need to decide an instance of subgraph isomorphism. □

4.7.2 A query complexity lower bound for the constructed property

Consider again the hard-to-test **PTIME**-decidable property Π^q constructed in Subsection 4.2.2. When used in Section 4.5, its query complexity (and its lack of isolated vertices) was sufficient for establishing a lower bound on testing its blowups. Our analysis here will have to be a bit finer, as we will not be using a reduction proper — neither from Π^q of Section 4.5 nor from Π'' of Subsection 4.2.2.

We recall that by Lemma 4.2.6, $\Omega\big(n^2\big)$ queries are required to distinguish between the distributions $\mathcal{G}_n$, a uniform distribution over graphs in Π''_n, and $\mathcal{R}_n$, a separating augmentation (as per Definition 4.2.2) of a uniformly sampled graph of order $\lfloor (n-1)/3 \rfloor$. Let us now carry this result over to dispersal-augmented graphs.

Before stating our lemma, we first note that our graphs of order $m(n,q)$ are now the results of dispersing augmentations. Recalling the definition of these augmentations (Definition 4.2.2), and denoting by $m'(n,q)$ the order of a pre-augmented graph, we have $m = m' + \frac{3}{2}2^{\lceil \log(2m'+1) \rceil}$. Since $m < 2^{1+\lceil \log(2m'+1) \rceil}$, we have $\lfloor \log(m) \rfloor = \lceil \log(2m'+1) \rceil$, so $m' = m - \frac{3}{2}2^{\lfloor \log(m) \rfloor}$.

Now, the indistinguishable distributions for Π^q are (for sufficiently large n):

$\mathcal{R}'_n$: A graph sampled from distribution $R_{m'(n,q)}$, dispersal-augmented to order $m(n,q)$, then blown up to order n.

G'_n: A graph sampled from distribution $G_{m'(n,q)}$, dispersal-augmented to order $m(n,q)$, then blown up to order n.

Lemma 4.7.5. *With $\mathcal{R}'_n$ and G'_n as per the above, any probabilistic oracle machine M making $o(q(n))$ queries to its input graph satisfies* $\mathbf{Pr}\Big[M^{\mathcal{R}'_n} = 1\Big] = \mathbf{Pr}\Big[M^{G'_n} = 1\Big]$.

Proof. We repeat an argument used in proving Lemma 4.2.6: As both distributions R'_n and G'_n are obtained by applying the same dispersing augmentation and blowup to samples from G_m and R_m respectively, and as the result of each query to a dispersing augmented graph depends on one or no edges of the original pre-augmented graph, and the result of each query to a blowup depends on one or no edges of the pre-blown-up graph — it suffices to prove the claim assuming queries are made to the original order-m graphs from R_m or G_m respectively — rather than to blowups of their dispersing augmentations. Lemma 4.2.6 proves, specifically, that if $o(q(n))$ queries are made, a machine has the same probability of accepting graphs from these two distributions. □

Lemma 4.7.6. *A graph sampled from R'_n is $c_{4.7.1}\delta_{4.2.5}/(250 \cdot 64)$-far from Π^q with probability $1 - o(1)$.*

Proof. By Lemma 4.2.5, with probability at least $1 - o(1)$ a graph G sampled from $R_{m'}$ is $\delta_{4.2.5}$-far from $\Pi''_{m'}$. Supposing this is the case, consider some graph $H \in \Pi''_{m'}$. By Lemma 4.5.6, the dispersing augmentation of G will be $\delta_{4.2.5}/250$-far from the dispersing augmentation of H. Now, the blowup of the dispersing augmentation of G to order n is a *balanced* blowup of an 1/8-dispersed graph, so by Lemma 4.7.1, it is $c_{4.7.1}(1/8)^2 \cdot \delta_{4.2.5}/250$-far from any generalized blowup of the dispersing augmentation of H. The claim follows when recalling that Π^q is the set of all generalized blowups of dispersing augmentations of graphs in $\Pi''_{m'}$. □

We can now prove the lower bound, setting $\varepsilon_{4.7} = c_{4.7.1}\delta_{4.2.5}/(250 \cdot 64)$:

Lemma 4.7.7. *Any ε-test for Π^q, for $\varepsilon \leq \varepsilon_{4.7}$, makes $\Omega(q(n))$ queries.*

Proof. Let n be sufficiently large for Lemma 4.7.5 to hold. For $\varepsilon \leq \varepsilon_{4.7}$, an ε-test for Π making less than $o(q(n))$ queries, which accepts graphs in Π with probability at least 2/3, will accept a graph sampled from $\mathcal{R}'_n$ with probability at least 2/3 (by Lemma 4.7.5). Now, by Lemma 4.7.6, with probability $1 - o(1)$, a graph from $\mathcal{R}'_n$ is $c_{4.7.1}\delta_{4.2.5}/(250 \cdot 64) = \varepsilon_{4.7}$-far from Π^q, so the probability of the test accepting a graph sampled from $\mathcal{R}'_n$ which are $\varepsilon_{4.7}$-far from Π^q is at least $2/3 - o(1)$. Thus for every sufficiently large n there exists a specific graph in the support of $\mathcal{R}'_n$ which is $\varepsilon_{4.7} > \varepsilon$-far from ${\Pi^q}_n$, and is accepted with probability over 1/2 — a contradiction. □

4.7.3 A one-sided test for the constructed property

Algorithm 4.5 will be the test achieving the upper bound.

Algorithm 4.5 A test for Π^q

1: Compute $m(n, q)$.
2: Uniformly sample a set S of $\Theta(m/\varepsilon)$ vertices.
3: Query the subgraph G_S induced by S.
4: If G_{sample} is a relaxed generalized blowup of a graph in Π''_m to order $|S|$, accept. Otherwise reject.

We stress that the test does not expect its sample to be a proper generalized blowup of a graph Π'' to order n — that is, it may include merely a subset of the clusters of such a blowup. Looking at the test, one may wonder whether it doesn't, perhaps, accept graphs which, despite being relaxed generalized blowups of graphs in Π'', are still far from proper generalized blowups (with all m clusters present); before proceeding to proving the test's validity, we allay this concern:

Lemma 4.7.8. *Suppose $n > 4m/\varepsilon$, and let G_1 and G_2 be graphs of order m and n respectively, such that G_2 is ε-far from any proper generalized blowup of G_1. Then G_2 is $\varepsilon/2$-far from any relaxed generalized blowup of G_1.*

Proof. Let G_1' be a relaxed generalized blowup of G_1 at minimum distance from G_2. To make G_1' into a proper generalized blowup of G_1, one must 'populate' the missing G_1-vertex clusters with vertices from other clusters, which now have more than one vertex. There are at most $m-1$ missing clusters, and moving a vertex from one cluster to another requires at most $n-1$ edge modifications. Thus the total number of edge modifications required to populate all clusters is less than $(n-1)\cdot(m-1) < (n-1)\cdot\left(\frac{\varepsilon n}{4}-1\right) < \frac{1}{2}\varepsilon\binom{n}{2}$, i.e. G_1' is $\varepsilon/2$-close to a proper generalized blowup of G_1. It must therefore be the case that G_1 and G_1' are at least $\varepsilon/2$-far. □

As the test samples more than $4m/\varepsilon$ vertices, the case of graphs with less than this many vertices is covered by the default behavior on graphs with too few vertices — querying the entire graph and deciding deterministically (see Definition 2.1.3 and the following discussion). It thus suffices if the test rejects graphs of higher order which are $\varepsilon/2$-far from relaxed generalized blowups of graphs in Π''.

Lemma 4.7.9. *Algorithm 4.5 is a valid test for Π^q with one-sided error, making $O(q(n))$ queries.*

Proof. Clearly, a graph G satisfying Π^q is accepted with probability 1, as G is particularly such a blowup, and being a relaxed generalized blowup of one of a set of graphs is hereditary: Losing vertices simply means having smaller clusters (due to the relaxation the clusters may be reduced to having 0 vertices).

Now suppose that G' is $\varepsilon/2$-far from ${\Pi^q}_n$; we assume without loss of generality that $n > 4m/\varepsilon$. Think of S as being sampled in $2m$ iterations, each adding $O(1/\varepsilon)$ newly-sampled vertices to S. Let S_i denote the sample in the i^{th} iteration and let $S^{\leq i} = \bigcup_{j\in[i]} S_j$. Consider $G_{S^{\leq i}}$, after the i^{th} iteration; suppose that it is a relaxed generalized blowup of a graph in ${\Pi''}_m$. In this case, Lemma 4.7.10 below guarantees that a uniformly sampled pair of vertices, when added to S, increases the number of clusters over the number in $G_{S^{\leq i+1}}$ with probability $\Omega(\varepsilon)$; when this pair is sampled from $V(G)\setminus S$, the probability can only be higher. Thus with probability at least $2/3$, at least one of the $O(1/\varepsilon)$ pairs increases the number of clusters. Consequently, over all m iterations, our sampled subgraph has probability at least $1-\exp(-\Omega(m)) > 2/3$ of being rejected either for reaching more than m clusters in the subgraph, or for having an induced subgraph which is itself not a relaxed generalized blowup of any graph in ${\Pi''}_m$, discovered already in an early iteration.

Finally, the number of queries is $\binom{\Theta(m(n,q)/\varepsilon)}{2} = \Theta\big(m^2/\varepsilon^2\big) = \Theta\big(q(n)/\varepsilon^2\big)$ (see Observation 4.5.10 regarding the last transition). Ignoring the dependence on ε, this is indeed $O(q(n))$. □

Lemma 4.7.10. *Let G be a graph of order $n > 2m\varepsilon$ which is ε-far from ${\Pi^q}_n$, and let $G_{S'}$ be the subgraph of G induced by a set of vertices $S' \subseteq V(G)$. Let m' denote the number of clusters in $G_{S'}$. Suppose that $m' \leq m(n,q)$ and that $G_{S'}$ is a relaxed*

generalized blowup of some graph in Π''_n*. Then for a uniformly sampled pair of vertices* $\{u', v'\}$*, there is a probability of at least* $\varepsilon/8$ *that* $S' \cup \{u, v\}$ *induces a graph with more than* m' *clusters.*

Proof. We first apply Lemma 4.7.8: Since G is ε-far from any proper generalized blowup of a graph in Π''_m, it is $\varepsilon/2$-far from any relaxed generalized blowup of a graph in Π''_m.

Now, let $G' \in \Pi''_m$ be the graph of which $G_{S'}$ is a relaxed generalized blowup. We note that, specifically, $G_{S'}$ is a proper generalized blowup of an induced subgraph G'' of G', with $|V(G'')| = m'$.

Consider a clustering of all vertices of G using S' as a signature, i.e. vertices with the same neighbors in S' are in the same cluster. Some of these clusters contain vertices from S' (let C_v denote the cluster containing $v \in S'$), and some may be *new*, with S' neighborhoods differing from all existing vertices in S'. If G has at least $\varepsilon n/8$ vertices in new clusters, one of them is sampled with probability at least $\varepsilon/8$, and the claim follows, since it will constitute a new cluster in the sampled subgraph.

If, on the other hand, there are few new clusters, the clustering is at "risk" of contradicting our assumption regarding G: If clusters C_u and C_v, for most pairs $u, v \in S'$, are mostly consistent with $\{u, v\}$ with respect to to the edge relation, then G can be made into a relaxed generalized blowup of G'' using few modifications. Specifically, it must be the case that at least $\frac{1}{4}\varepsilon\binom{n}{2}$ edges $\{u', v'\}$ with $u' \in C_u$ and $v' \in C_v$ for the corresponding $u, v \in S'$ have $\{u', v'\} \in E(G'')$ iff $\{u, v\} \notin E(G'')$; otherwise one can correct all these discrepancies, then move all new-cluster vertices to S'-vertex clusters, with at most $n-1$ edge changes per vertex, for a total of $\frac{1}{8}\varepsilon(n-1)n = \frac{1}{4}\varepsilon\binom{n}{2}$ additional changes.

Consequently, when sampling two new additional vertices u', v' from S' clusters (denoted C_u and C_v), with probability at least $\varepsilon/4$ we find that they do not agree with their cluster with respect to being an edge. It must then be the case that the number of clusters in $G_{S' \cup \{u,v\}}$ must increase when clustering according to the neighborhoods in $S' \cup \{u, v\}$. □

Theorem 4.7 is now proven by a combination of the query complexity lower bound of Lemma 4.7.7, the upper bound established through the valid test in Lemma 4.7.9, and Lemma 4.7.4 regarding the decidability of Π^q.

Chapter 5

Lower bounds for testing partite dense structures

5.1 Introduction and overview of results

While testing graphs has received the most attention in the research of combinatorial property testing (specifically, testing graphs in the dense model), other dense structures are also of interest. Some are strictly more expressive than graphs (see the discussion of hypergraph partition properties in Chapter 6 below), some strictly less expressive, such as bipartite graphs, and some have both restrictions and extensions of the power of expression. This chapter considers the latter case: Bipartite graphs, but with edges in multiple colors; and k-uniform hypergraphs which are also k-partite (referred to as *k-graphs* for short throughout this section).

For strictly less expressive structures — in the same testing model essentially, the dense model in our case — upper bounds on testing more expressive structures generally apply, while lower bound results for more expressive structures come into question, as one may expect to provide stronger upper bounds by exploiting the structural restrictions.

Such expectation was indeed justified for the case of bipartite graphs, with properties defined by a family of forbidden subgraphs. In general graphs, testing arbitrary such properties (without relying on the size of the input graph) requires the use of Szemerédi 's regularity lemma, resulting in extremely poor dependence of the query complexity on ε. While the known lower bounds are not at all close to the tower functions incurred by the use of regularity, they are super-polynomial, and there is certainly reason to suppose that the minimum query complexity of such tests is in fact much higher. As it turns out, in bipartite graphs this is not the case.

Fischer and Newman showed, in [FN01], a first upper bound for testing forbidden induced subgraphs in bipartite graphs (viewing them as binary matrices, see below) — although this was doubly-exponential in $1/\varepsilon$ and was not known to contradict the established lower bounds, it was a non-regularity based technique, that could not be applied as such to general graphs. It was improved upon in Alon, Fischer and Newman's

[AFN07], yielding a *polynomial* upper bound on such tests. Interestingly, the main tool of [AFN07] is a sort of a regularity lemma — but with a conditional: Either a bipartite graph has a relatively small "regular" partition (we shall not go into the details of the definition of regularity here) of size polynomial in the regularity parameter, or every possible small induced subgraph appears in the graph, in significantly many copies. In the former case one can construct a 'signature' of the regular partition using polynomially many queries, and decide based on this signature; in the latter case, a uniformly-sampled subgraph will, with high probability, contain a forbidden structure.

One would hope that such a technique may apply in somewhat more general contexts: Instead of just bipartite graphs, graphs with edge colors (or alternatively, matrices over a finite domain larger than $\{0,1\}$); or in higher dimension — k-edges instead of two-edges, k-partite hypergraphs instead of bi-partite graphs (or alternatively, binary tensors instead of matrices). This was an open question posed in [AFN07].

After some efforts attempting to extend the upper bound further, research has yielded the opposite — an establishment of lower bounds, precluding this possibility:

Theorem 5.1. *There exists a 2-colored bipartite graph F with two vertices per part, such that any ε-test of 3-colored bipartite graphs for being free of having F as an induced subgraph, performs no less than $(c/\varepsilon)^{c\cdot\ln(c/\varepsilon)}$ queries for some global constant c.*

Theorem 5.2. *There exists a 3-uniform tripartite hypergraph F with two vertices in each part, such that every ε-tester of 3-uniform tripartite hypergraphs for being free of copies of F, as an induced sub-hypergraph, performs no less than $(c/\varepsilon)^{c\cdot\ln(c/\varepsilon)}$ queries for some global constant c.*

The proofs use constructions based on adaptations of the lower bound known for testing forbidden subgraphs (specifically, triangles) for general graphs.

5.2 Additional preliminaries

The following table summarizes the specifics of dense model testing, for colored bipartite graphs and for k-graphs, in comparison with the case of (general, non-partite) graphs:

Structures	**Graphs**	**σ-Colored (Complete) Bipartite Graphs**	**k-Graphs**
query	"$\{x,y\} \in E$?"	"what is $\mathbf{col}(x,y)$?" with **col** values ranging in $\{0,\ldots,\sigma-1\}$	"$(x_1,\ldots,x_k) \in E$?"
maximum distance between structures	$\binom{n}{2}$	n^2	n^k
vertex set(s)	V	U, V	$V_1,\ldots,V_k$
meaning of the order n	$\lvert V\rvert$	$\lvert U\rvert = \lvert V\rvert$	$\lvert V_1\rvert = \ldots = \lvert V_k\rvert$

A matrix M over $\{0,1\}$ can be associated with the (labeled) bipartite graph $G = (U, V, E)$, with

$$\begin{aligned} U &= \{u_1, \ldots, u_n\} \\ V &= \{v_1, \ldots, v_n\} \\ E(G) &= \{(u_i, v_j) \in U \times V \mid M(i,j) = 1\} \end{aligned}$$

that is, the bipartite graph whose adjacency matrix is M. Similarly, a matrix over a larger domain $\{0, \ldots, \sigma - 1\}$ can be associated with an appropriate σ-colored bipartite graph; and a k-dimensional tensor T over $\{0,1\}$ can be associated with the k-graph of which it constitutes the adjacency tensor.

The conceptual similarity between matrices or tensors, and representations of colored bipartite graphs or of k-graphs, will be used implicitly throughout this chapter. Note, however that properties are closed under relabeling, i.e. a permutation of the indices on the axes of the matrix/tensor in all dimensions. Thus, when we refer to 'submatrices' of a bipartite graph's adjacency matrix, we are actually referring to subgraphs — the submatrix coordinates may be selected irrespectively of the order of coordinates in the adjacency matrix.

Finally, our lower bound constructions also involve the following:

Definition 5.2.1. A *cyclic* k-partite digraph $G = (V_1, \ldots, V_k, E)$ is a k-partite digraph in which every edge in E extends from V_i to $V_{(i \bmod k)+1}$ for some $i \in [k]$.

5.3 A lower bound for colored bipartite graphs

Our proof for Theorem 5.1 will be based on the argument that any test (not just tests with one-sided error) must, in some sense, find copies of forbidden subgraphs; see the discussion in Section 3.3, and specifically, Corollary 3.3. We will thus be proving the following key lemma, regarding the scarcity of forbidden subgraphs:

Lemma 5.3.1. *There exists a* $(2,2)$ *bipartite graph* F*, such that for every* ε *and for every* $n > 16(c/\varepsilon)^{-c \cdot \ln(c/\varepsilon)}$*, there exists a 3-colored bipartite graph* G *which is* ε*-far from being free of* F*, while the fraction of* $(2,2)$ *subgraphs of* G *which are copies of* F *is at most* $(c/\varepsilon)^{-c \cdot \ln(c/\varepsilon)}$*, for some global constant* c*.*

In leading up to a proof of this lemma will shall begin with a simple and rough construction: Describing how the adjacency matrix of a colored bipartite graph can represent partite cyclic digraphs, with the representation preserving the distributions of induced substructures in the digraph; we shall then construct 4-partite cyclic digraphs in which induced directed 4-cycles are super-polynomially rare.

Such a construction will prove a weaker version of Lemma 5.3.1: For one, we will have used many more than 3 colors — the representation of a digraph will not be very

terse; also, we will have used numerous forbidden submatrices, as a submatrix of the digraph representation will also contain information about additional edges to those constituting a 4-cycle. We will then proceed to make several refinements:

1. One may construct digraphs with few 4-cycles as described above, with the additional constraint that the first three edge layers are identical.

2. One may construct 4-partite digraphs as described in item 1, with the additional constraint that the edge layers are symmetric with respect to a relevant ordering of the vertices in each part.

3. The construction for item 2 can be shown to satisfy the additional constraint that no pair of vertex indices is such that its corresponding pairs of vertices are connected in all four edge layers.

These successive refinements will bring every pair of vertex indices j_1, j_2 to have only three possible edge configurations; consequently, we will only need three colors for the matrix representation of the digraph, and only one forbidden submatrix (i.e. only one forbidden subgraph).

5.3.1 Representing cyclic partite digraphs by matrices

Given a cyclic k'-partite digraph, we decompose its edges into k' bipartite digraphs between pairs of cycle-consecutive parts. The edge relation between each of these pairs can be thought of in terms of its binary adjacency matrix, leading to the following representation:

Definition 5.3.2. Let $G = (V_1, \ldots, V_{k'}, E)$ be a cyclic k'-partite digraph, with k' vertex sets of size n each, where V_i=$(v_{i,1}, \ldots, v_{i,n})$. The *matrix representation of* G, denoted $\mathcal{CM}(G)$, is the matrix of order n, over a domain of size $2^{2k'}$ (the set of *cell colors*), corresponding to all possible combinations of the following $2k'$ binary values: For $M = \mathcal{CM}(G)$, each cell $M(j_1, j_2)$ has a distinct color bit for each one of the k' edges $(v_{1,j_1}, v_{2,j_2}), \ldots, \left(v_{k'-1,j_1}, v_{k',j_2}\right), \left(v_{k',j_1}, v_{1,j_2}\right)$, and another bit for each one of the k' edges $(v_{1,j_2}, v_{2,j_1}), \ldots, \left(v_{k'-1,j_2}, v_{k',j_1}\right), \left(v_{k',j_2}, v_{1,j_1}\right)$. Each bit is set to 1 if its respective edge exists, and to 0 otherwise.

Our lower bound construction utilizes cyclic 4-partite digraphs which are far from not containing a (directed) 4-cycle, yet have few copies of it; we consequently set henceforth $k' = 4$. The reason for this choice of the number of parts is that 4 is the lowest even number of parts with an induced subgraph for which testing freeness is hard — as described in Alon and Shapira's [AS04b]. Our matrix representations $\mathcal{CM}(\cdot)$ therefore has cells with $2^{2k'} = 2^8 = 256$ possible values.

The forbidden submatrices

Querying a matrix cell at (j_1, j_2) yields information about edges in all four layers; querying a 2×2 submatrix with coordinates $(j_1, j_3) \times (j_2, j_4)$ yields information about several directed 4-cycles, one of which is $C = (v_{1,j_1}, v_{2,j_2}, v_{3,j_3}, v_{4,j_4})$. We note that for every 4-cycle of G there is a choice of $j_1, \ldots, j_4$ for which C as defined above corresponds to that cycle. We thus only need to forbid 2×2 submatrices witnessing the existence of the four edges of the single directed cycle C associated with a given submatrix. There are many possible such 2×2 submatrices, as the existence of any of the rest of the $(k'/2)^2 \cdot 2k' - k' = 28$ edges represented in the submatrix cells does not affect the presence of C. The forbidden submatrices are therefore the 2^{28} matrices in which the four color bits for the edges of C are set.

Note that in some cycles of G it may be the case that $j_1 = j_3$ and/or $j_2 = j_4$. We refer to such cycles as *degenerate*; our construction and our arguments below will only involve graphs with no degenerate cycles, so we may disregard these.

For every copy of a (non-degenerate) 4-cycle in G, there exists exactly one order-2 forbidden submatrix in $\mathcal{CM}(G)$ (recall that the submatrix may appear permuted in rows or columns). This is true despite the fact that it is possible to *infer* the existence of a 4-cycle also from other submatrices of $\mathcal{CM}(G)$. In other words, a selection of a order-2 submatrix of $\mathcal{CM}(G)$, and a check of whether its C exists, correspond to a selection of four vertices in the four parts of G and a check of whether they form a (non-degenerate) cycle. With $n = |V_i|$ as the size of each V_i, There are $(n(n-1))^2$ such possible choices.

5.3.2 An initial hard-to-test matrix

Definition 5.3.3. The *trivial* integer solutions to the equation $x_1 + x_2 + \ldots + x_r = r \cdot x_{r+1}$ are those in which all of $x_1, \ldots, x_r$ are equal.

Lemma 5.3.4 ([Alo02, Lemma 3.1] and [AS04b, Lemma 6.1])**.** *For every natural $r \geq 2$, and for every positive integer m, there exists a subset $X_m \subseteq [m]$, of size at least $\exp\left(-10\sqrt{\ln(m)\ln(r)}\right) \cdot m$, with no non-trivial solution to the equation $x_1 + x_2 + \ldots + x_r = r \cdot x_{r+1}$.*

Fix $r = 3$ and $\varepsilon' = 8\varepsilon$. Let m be the maximum possible satisfying $\varepsilon' m < 7 \cdot 2^{-1} \cdot 4^{-4}|X_m|$, obtaining, for an appropriate constant c, the bound $m \geq (c/\varepsilon')^{c \cdot \ln(c/\varepsilon')}$.

Using such a set $X = X_m$, we construct a cyclic 4-partite digraph T: The four parts of T's vertex set, $V_1, \ldots, V_4$, have cardinalities $m, 2m, 3m, 4m$ respectively. For every $i \in \{1, 2, 3\}$, $j \in [im]$ and $x \in X$, T has the edge (v_j, v_{j+x}) between V_i and V_{i+1}; T also has the edges between V_4 and V_1 of the form (v_{j+3x}, v_j), for every $x \in X$ and $j \in [m]$.

As one may verify (see [AS04b, Lemma 6.2]), $E(T)$ contains $m|X|$ edge-disjoint copies of the directed 4-cycle, formed by $4m|X|$ edges, and no other directed 4-cycles; T's total number of edges is $(1 + 2 + 3 + 1) \cdot |X| \cdot m > 2 \cdot 4^4 \varepsilon' m^2$. For our purposes we would like all parts V_i to have the same size, so we add isolated vertices making every

V_i of size exactly $4m$. Let T_1 be the graph resulting from this addition. We note that all cycles of T_1 are non-degenerate.

Lemma 5.3.5 (special case of [AS04b, Lemma 6.3])**.** *Let $K = (V(K), E(K))$ be a digraph and let $T = (V(T), E(T))$ be an s-factor blowup of K. Let R be a subset of the set of edges of T, and suppose that each copy of K in T contains at least one edge of R. Then $|R| > |E(T)|/|E(K)|^2 > |E(T)|/|V(K)|^4$.*

Now let G_1 denote an s-factor blowup of T_1, with $s = \lfloor n/(4m) \rfloor$. We have $|E(G_1)| \geq s^2 \cdot 2 \cdot 4^4 \varepsilon' m^2 > 4^2 \varepsilon' n^2$. Since $E(T_1)$ consists of edge-disjoint 4-cycles, $E(G_1)$ consists of edge-disjoint s-blown-up 4-cycles. By Lemma 5.3.5, with a 4-cycle as K, at least a $\frac{1}{|E(K)|^2} = \frac{1}{4^2}$-fraction of the edges of each of these s-blown-up 4-cycles must be removed so as to remove all 4-cycles from G_1; G_1 is thus ε'-far from being 4-cycle-free. On the other hand, as $|X| \leq m$, G_1 has $m|X| \cdot s^4 \leq m^2 s^4 < 4^4 n^4/m^2$ copies of the 4-cycle. One can also verify that all cycles of G_1 are non-degenerate.

We must now transform the argument regarding the scarcity of forbidden subgraphs in G_1 to apply to forbidden submatrices in $\mathcal{CM}(G_1)$.

Proposition 5.3.6. *For $\sigma = 2^8$ there exists a finite set $\mathcal{F}$ of σ-colored order-2 matrices, such that for every ε and $n > (c/\varepsilon)^{c \cdot \ln(c/\varepsilon)}$, there exists a σ-colored matrix M which is ε-far from being free of members of $\mathcal{F}$, and yet, the fraction of order-2 submatrices of M which are copies of a member of $\mathcal{F}$ is at most $(c/\varepsilon)^{-c \cdot \ln(c/\varepsilon)}$ for some global constant c.*

Proof. Let $M = \mathcal{CM}(G_1)$, and set the family of forbidden matrices to be the 2^{28} matrices defined above.

To prove the second part of the claim we recall that there is only one copy of a forbidden matrix in $\mathcal{CM}(G_1)$ for every copy of a 4-cycle in G. Only $c_1 n^4/m^2$ of the $(n(n-1))^2$ possible directed non-degenerate 4-cycles with vertices in consecutive parts appear in G, so no more than an $8c_1/m^2$ fraction of the $(n(n-1))^2$ submatrices of $\mathcal{CM}(G_1)$ of order 2 are copies of forbidden matrices.

For the first part of the claim, we note that by modifying a matrix cell one affects the representation of at most 8 edges of G_1. Thus, unless at least $\varepsilon' n^2/8 = \varepsilon n^2$ cells are modified, more than $(1-\varepsilon')n^2$ of the edges of G_1 have their two representing color bits (i.e. in both the cells $\mathcal{CM}(G_1)(i,j)$ and $\mathcal{CM}(G_1)(j,i)$) unmodified. In this case, G_1 still has a 4-cycle with its representing order-2 submatrix intact, i.e. $\mathcal{CM}(G_1)$ still contains a copy of a forbidden matrix. □

5.3.3 Reducing the number of colors

As mentioned above, 256 colors are more than necessary to construct a hard to test matrix. We now reduce this number by refining our construction; as we do so, we lose the expressivity of matrices; we maintain, however, the ability to represent the particular graphs we construct for proving the lower bound.

Making most edge layers identical

We note that in the graph T_1, the edge sets in the 'first' 3 layers, those between V_i and V_{i+1} for $1 \leq i \leq 3$, are quite similar: $v_{i,j}$ is connected to $v_{i,j+x}$. The difference is that in each of the V_i's, only the first im vertices are connected onwards to vertices in V_{i+1}. We now add "$+x$" edges for all vertices in each V_i, not just the first im vertices — that is, we make $(v_{i,j}, v_{i+1,j+x})$ an edge whenever $j + x \leq 4m$ and $x \in X$.

Let T_2 denote this new graph. As with the graph T_1, every directed 4-cycle $(v_{1,j_1}, v_{2,j_2}, v_{3,j_3}, v_{4,j_4})$ in T_2 satisfies

$$(j_2 - j_1) + (j_3 - j_2) + (j_4 - j_3) = (j_4 - j_1)$$

so when denoting

$$x_1 = j_2 - j_1 \qquad x_2 = j_3 - j_2 \qquad x_3 = j_4 - j_3 \qquad x_4 = (j_4 - j_1)/3$$

the equation becomes $x_1 + x_2 + x_3 = 3x_4$; since $x_1, \ldots, x_4 \in X$, all four x values must be equal. Also, if such a cycle begins with $j_1 > m$ in V_1, then

$$(j_1 - m\lfloor j_1/m \rfloor, j_2 - m\lfloor j_1/m \rfloor, j_3 - m\lfloor j_1/m \rfloor, j_4 - m\lfloor j_1/m \rfloor)$$

is another cycle in T_2 (the vertex indices all remain positive), which begins with $j_1 \leq m$, i.e. it corresponds to a cycle in the original T_1. It follows that the total number of cycles has increased by no more than a factor of 4, and that all cycles are still non-degenerate.

Since all cycles are edge-disjoint in T_2 as well, the number of cycles increases with the s-factor blowup of T_2 into G_2 by a factor of s^4, as in the case of G_1. G_2 has the same vertex sets as G_1, and a superset of the edges of G_1, making it at least as far from being 4-cycle free as G_1. As for the number of cycles, T_1 had at most $m|X| < m^2$ 4-cycles, T_2 has at most $4m|X| < 4m^2$ 4-cycles, and G_2 has at most $4m^2 s^4 < c_2 n^4/m^2$ 4-cycles, for some constant c_2.

We can now use our different construction of T_2 to reduce the number of colors necessary for its representation: As the bits for the three $V_i \to V_{i+1}$ edge layers are the same, we only need two bits for each type of layer (one for the $j_1 \to j_2$ edge and one for the 'flip' edge $j_2 \to j_1$), times two types of layers ($V_i \to V_{i+1}$ and $V_4 \to V_1$): in total we now use only $2^4 = 16$ colors. This property of T_2's first three layers carries over to G_2 with the blowup.

Our observations thus lead us to conclude that Proposition 5.3.6 also holds for $\sigma = 2^4$, with a different choice of the constants.

Making the edge layers symmetric

The number of color bits may be further reduced — halved — if we ensure that whenever $(v_{i_1,j_1}, v_{i_2,j_2})$ is an edge, so is $(v_{i_1,j_2}, v_{i_2,j_1})$. To achieve this, we could add the 'flip' edges

to T_2 – in addition to the edge (j_1, j_1+x) between V_i and V_{i+1}, and the edge (j_1+3x, j_1) between V_4 and V_1, we could add (j_1+x, j_1) between V_i and V_{i+1}, and (j_1, j_1+3x) between V_4 and V_1 respectively.

The addition of the 'flip' edges may, however, result in an excessive increase in the number of cycles, and possibly also result in intersections of the edges of different cycles. To avoid this, we again modify our pre-blowup graph T. Let us first consider a replacement of T_2 by the following T_3': Each of the four vertex sets is now $\{1, \ldots, 4^2 \cdot m\}$. The edges in the first three layers (which continue to be uniform) are (j_1, j_1+x+3m) for all $x \in X$ and $j_1 \in [4^2m - x - 3m]$; the edges between V_4 and V_1 are $(j_1+3(x+3m), j_1)$ for all $x \in X$ and $j_1 \in [4^2m - 3(x+3m)]$. Each directed 4-cycle $(v_{1,j_1}, v_{2,j_2}, v_{3,j_3}, v_{4,j_4})$ must still satisfy

$$(j_2 - j_1) + (j_3 - j_2) + (j_4 - j_3) = (j_4 - j_1)$$

We denote

$$x_1 = j_2 - j_1 - 3m \quad x_2 = j_3 - j_2 - 3m \quad x_3 = j_4 - j_3 - 3m \quad x_4 = (j_4 - j_1 - 9m)/3$$

and this yields again the equation $x_1 + x_2 + x_3 = 3x_4$. Thus as in the case of T_2 above, cycles only exist when the edge x-values are all equal, i.e. T_3' has no more than $4^2m|X|$ copies of a 4-cycle.

We now add all flip edges to T_3': the edges of the form $(v_{i,j_1+x+3m}, v_{i+1,j_1})$ are added in the first three layers, and the edges of the form $\left(v_{4,j_1}, v_{1,j_1+3(x+3m)}\right)$ are added in the fourth layer. Let T_3 denote the resulting graph.

Lemma 5.3.7. *Every cycle in T_3 is either a cycle in T_3' (a no-flip-edge cycle) or a cycle consisting only of flip edges.*

Proof. Consider first some tuple (j_1, j_2, j_3, j_4) of vertex indices in the four parts where the first two edges are non-flip while the third one is a flip edge. In this case, we find that j_4 cannot be very far from j_1:

$$|j_1 - j_4| = |(j_1 - j_2) + (j_3 - j_2) - (j_3 - j_4)| \leq 2 \cdot (3m + m) - 3m < 9m$$

however, for (j_4, j_1) to be an edge in the fourth layer (either a non-flip or a flip edge), we must have $|j_4 - j_1| = 9m + 3x$ for some $x \in X$. No such edges exist, proving that such a cycle is impossible. The remaining cases where one of three $V_i \to V_{i+1}$ edges is in the direction opposite to the other two edges are similarly impossible, implying that the edges in the first three layers are in the same 'direction' for every cycle of T_3. If these three edges are non-flip edges, the j's are an increasing sequence, and so the fourth edge must have $j_4 > j_1$, i.e. it must also be a non-flip; if the edges in first three layers are flip edges, the j's are a decreasing sequence, and $j_4 < j_1$, i.e. the fourth edge must also be a flip edge. □

As for the number of cycles with all-flip or all-non-flip edges: If $(v_{i_1,j_1}, v_{i_2,j_2})$ is a non-flip edge, then $\left(v_{i_1,4^2m-j_1+1}, v_{i_2,4^2m-j_2+1}\right)$ is a flip edge, and $\left(v_{i_1,4^2m-j_2+1}, v_{i_2,4^2m-j_1+1}\right)$ is a non-flip edge. Thus if

$$(v_{1,j_1}, v_{2,j_2}, v_{3,j_3}, v_{4,j_4})$$

is a cycle with no flip edges, then

$$\left(v_{1,4^2m-j_1+1}, v_{2,4^2m-j_2+1}, v_{3,4^2m-j_3+1}, v_{4,4^2m-j_4+1}\right)$$

is an all-flip-edge cycle, and vice-versa. This bijective correspondence, together with the lemma above, bring us to conclude that there are exactly twice as many cycles in T_3 as there are in T_3', and that they are all edge-disjoint. Furthermore, the necessity of the first three edges to be in the same direction means that $j_1 \neq j_3$ and $j_2 \neq j_4$, so all cycles are still non-degenerate.

T_3 is a graph with 4^2m vertices in each part and no more than $2 \cdot 4^2m|X|$ 4-cycles, all edge-disjoint. Blowing it up by a factor of $s = n/\left(4^2m\right)$, we obtain a graph G_3 with n vertices per part and $2 \cdot 4^2m|X| \cdot s^4 \leq c_3 n^4/m^2$ cycles for an appropriate constant c_3. G_3 is also ε'-far from being cycle-free, by an argument similar to the case of G_1, with a proper choice of $m(\varepsilon') = (c/\varepsilon)^{c \cdot \ln(c/\varepsilon)}$ reflecting the change in the constants used in the construction of T_3 and the blowup.

To represent G_3, we only need two bits of color: One bit for the first three layers (a single bit now suffices for both the 'non-flip' and the 'flip' edge), and one bit for the $V_4 \to V_1$ layer. We have thus brought down σ, the domain size for matrix cell values for which Proposition 5.3.6 holds, to $2^2 = 4$ (again, with a different choice of a constant c).

Mutual exclusion between the edge layers

How can we further reduce the number of colors? The upper bound result of [AFN07] implies that it is impossible to reduce the number of bits per cell from two to one, without making the matrix easy to test for the presence of forbidden submatrices. Still, a decrease from four to three colors is possible. In fact, if we review the construction of T_3 and G_3 carefully, we find that for any (j_1, j_2), we only have three edge combinations represented for (j_1, j_2) (and the now-symmetric (j_2, j_1)):

1. (j_1, j_2) is an edge in the $V_i \to V_{i+1}$ layers, but not in $V_4 \to V_1$.
2. (j_1, j_2) is not an edge in $V_i \to V_{i+1}$ layers, but is an edge in $V_4 \to V_1$.
3. (j_1, j_2) is not an edge in any layer.

No (j_1, j_2) can be an edge in all four layers, since edges in $V_4 \to V_1$ correspond to index differences $|j_1 - j_2|$ of at least $9m + 1$ (before the blowup of T_3 into G_3), while edges $V_i \to V_{i+1}$ correspond to differences of at most $4m$. Thus Proposition 5.3.6 holds for $\mathcal{CM}(G_3)$ as a 3-colored matrix as well. In fact, we are now able to prove Lemma 5.3.1:

Proof of Lemma 5.3.1. G is $\mathcal{CM}(G_3)$ constructed above. Indeed, there is now only one possible order-2 submatrix (up to permutations) of $\mathcal{CM}(G_3)$ witnessing the presence of its corresponding cycle C in G_3: $M_{\mathrm{F}} = \left(\begin{smallmatrix} 1 & 2 \\ 1 & 1 \end{smallmatrix}\right)$ (this is a matrix over $\{0,1,2\}$). One may verify that in all other order-2 submatrices, at least one of the cycle edges must be missing. Thus F is the subgraph with adjacency matrix M_{F}. □

5.3.4 Proof of the lower bound

Observation 5.3.8. The property of colored bipartite graphs being free of the forbidden subgraph F, of the proof of Lemma 5.3.1, is hereditary — like any property of being free of forbidden induced substructures. It is also inflatable, as F is not a blowup of a smaller graph.

With our construction in Lemma 5.3.1 and the above observation, we can now proceed to proving the lower bound theorem. As our proof makes use of the general results regarding dense structures (in Section 3.9), we make a final observation regarding the testing model to justify this use:

Observation 5.3.9. 3-colored bipartite graphs can be expressed as a class of general dense structures (as per Subsection 2.1.1): Two vertex sets $V_1 = U$, $V_2 = V$, and two edge relations E_1, E_2, with appropriate constraints. The constraints would be: Every edge of any of the two relations has the first vertex in U, and the second in V; and whenever $E_1(u,v)$ is an edge, $E_2(u,v)$ can't be an edge. A query of an edge of the 3-colored bipartite graph corresponds to two queries, of the values of E_1 and E_2, for the appropriate tuple. Also, the bipartite graphs we consider are of uniform order — both parts have the same number of vertices.

Proof of Theorem 5.1. Consider an ε-test of 3-colored bipartite graphs for being free of the forbidden subgraph F, which makes at most $q(\varepsilon)$ queries. As this property is hereditary and inflatable, we may apply Corollary 3.9 to this test, which is specifically a uniform-order test, to obtain a perfectly canonical one-sided test for F-freeness with queried subgraph order $q'(\varepsilon) \in \mathrm{poly}(q(\varepsilon))$.

By Lemma 5.3.1, there exists (for any sufficiently high n) a graph G and a forbidden subgraph F, such that G is ε-far from being free of F, but only a $(c'/\varepsilon)^{-c' \cdot \ln(c'/\varepsilon)}$ fraction of its order-2 subgraphs are copies of F, for some global constant c'.

The expected number of copies of F in the subgraph of G queried by the perfectly canonical test is no more than $\mathrm{O}\left(q'^4\right) \Big/ (c'/\varepsilon)^{c' \cdot \ln(c'/\varepsilon)}$ — the expected number of copies of $\mathcal{CM}(F)$ in a submatrix of $\mathcal{CM}(G)$ of order $O(q')$. Thus if $q(\varepsilon) < (c/\varepsilon)^{c \cdot \ln(c/\varepsilon)}$, for an appropriate constant c, then the expected number of forbidden subgraphs discovered is $o(1)$, so the test accepts G with probability $1 - o(1)$ — a contradiction. □

5.4 A lower bound for k-uniform k-partite hypergraphs

5.4.1 A hard-to-test tensor

Fix ε. Let M be as in the proof of Lemma 5.3.1, but with distance parameter $\varepsilon' = 2\varepsilon$. Let us again think of the 3-colored matrix M as having two color bit layers: One bit-layer for the first three edge-layers of the 4-cycle (V_i to V_{i+1}), and another bit-layer for the 4th edge-layer (V_4 to V_1); it is still the case that no matrix cell $M(j_1, j_2)$ has both of its bits set.

Let us separate M into two binary matrices M' and M'', with $M'(j_1, j_2)$ being the first color bit of $M(j_1, j_2)$ and $M''(j_1, j_2)$ being the second color bit. Using these two matrices, we construct a 3-dimensional tensor T of order n:

$$T(x, y, z) = \begin{cases} M'(x, y) & 1 \leq z \leq n/2 \\ M''(x, y) & n/2 < z \leq n \end{cases}$$

We split the forbidden order-2 matrix M_{F} of Lemma 5.3.1 in a similar fashion, to obtain a forbidden order-2 subtensor T_{F}:

$$\left[\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\right]$$

(the two matrices are the layers for the two values in the z coordinate).

Lemma 5.4.1. *Let T' be a subtensor of T with coordinates $(j_1, j_3) \times (j_2, j_4) \times (z_1, z_2)$. $T' = T_F$ if and only if the following holds: (j_1, j_2, j_3, j_4) are vertex indices of a cycle in G_3, $z_1 \in \left\{1, \ldots, \frac{n}{2}\right\}$ and $z_2 \in \left\{\frac{n}{2} + 1, \ldots, n\right\}$.*

Proof. If $z_1, z_2 \leq \frac{n}{2}$ or $z_1, z_2 > \frac{n}{2}$, then T' is invariant along the z-axis and is therefore not a copy of T_{F}. Now suppose that $z_2 \in \left\{1, \ldots, \frac{n}{2}\right\}$ and $z_1 \in \left\{\frac{n}{2} + 1, \ldots, n\right\}$; in this case, all of (v_{j_1}, v_{j_2}), (v_{j_3}, v_{j_2}) and (v_{j_3}, v_{j_4}) are edges in the fourth edge layer of G_3 and (v_{j_4}, v_{j_1}) is an edge in the first three edge layers. We recall that G_3 is a blowup of the graph T_3, thus there exist vertices $v_{j'_1}, \ldots, v_{j'_4} \in T_3$ such that $\left(v_{j'_1}, v_{j'_2}\right)$, $\left(v_{j'_3}, v_{j'_2}\right)$ and $\left(v_{j'_3}, v_{j'_4}\right)$ are edges in T_3's fourth edge layer, and $\left(v_{j'_4}, v_{j'_1}\right)$ an edge in its first three edge layers. Now, the edges in the fourth layer correspond to index differences $|j'_1 - j'_2|$,$|j'_3 - j'_2|$ and $|j'_3 - j'_4|$ of at least $9m + 1$. Thus either $j_1 < 5m$ or $j_1 > 9m + 1$. In the first case, $j_2 > 9m + 1$, $j_3 < 5m$ and $j_4 > 9m + 1$, thus $|j'_1 - j'_4| > 4m$, which makes it impossible for (j'_4, j'_1) to be an edge in the first three layers. The second case is similar. Thus whenever $z_2 \in \left\{1, \ldots, \frac{n}{2}\right\}$ and $z_1 \in \left\{\frac{n}{2} + 1, \ldots, n\right\}$, it is impossible that $T' = T_{\mathrm{F}}$.

Finally, suppose $(z_1, z_2) \in \left\{1, \ldots, \frac{n}{2}\right\} \times \left\{\frac{n}{2} + 1, \ldots, n\right\}$. In this case $T'(\cdot, \cdot, z_1)$ is the first color bit of a order-2 submatrix of M, and $T'(\cdot, \cdot, z_2)$ is the second color bit thereof. If (j_1, j_2, j_3, j_4) are not vertex indices of a cycle of G_4, then at least one of the four '1' bits of T_{F} must be missing from T', so again $T' \neq T_{\mathrm{F}}$.

For the second direction of the lemma, let (j_1, j_2, j_3, j_4) be vertex indices of a cycle in G_3, and let $(z_1, z_2) \in \{1, \ldots, \frac{n}{2}\} \times \{\frac{n}{2} + 1, \ldots, n\}$. The existence of the cycle constrains the four subtensor cells corresponding to the four edges to be 1, and the fact that no edge can exist both in the first three edge layers of G_3 and in its fourth layer constrains the other four bits to 0, so indeed $T' = T_{\mathrm{F}}$. □

5.4.2 Proof of the lower bound

Lemma 5.4.2. *There exists a single 3-dimensional binary tensor T_F of order 2, such that for every n, ε there exists a tensor T, which is ε-far from being free of T_F, yet the fraction of order-2 subtensors of T which are copies of T_F is at most $(c/\varepsilon)^{-c \cdot \ln(c/\varepsilon)}$, for some global constant c.*

Proof. Let T, T_{F}, M, M', M'' and ε' be as in Subsection 5.4.1. Lemma 5.4.1 ensures that for every choice of z-axis coordinates z_1, z_2, either no choices of $(j_1, j_3) \times (j_2, j_4)$ yield a copy of T_{F} (for the case of $z_1 > \frac{n}{2}$ or $z_2 \leq \frac{n}{2}$), or at most a $(c'/\varepsilon')^{-c' \cdot \ln(c'/\varepsilon')}$ fraction of these choices yield such a copy (due to the properties of M). Setting $c = c'/2$ we conclude that at most a $\frac{1}{4} \cdot (c'/\varepsilon')^{-c' \cdot \ln(c'/\varepsilon')} < (c/\varepsilon)^{-c \cdot \ln(c/\varepsilon)}$ fraction of the order-2 subtensors of T are copies of the forbidden subtensor.

As for the distance from being T_{F}-free, for every $z_1 \in [n/2]$, one must modify enough cells of $T(\cdot, \cdot, z_1) = M'$ and $T(\cdot, \cdot, z_1 + \frac{n}{2}) = M''$ to affect all copies of T_{F} located in this pair of layers. These copies are in bijective correspondence with the copies of the forbidden order-2 matrix in M, and the number of x, y coordinate pairs in which M has to be changed to remove all copies of the forbidden submatrix is at least $\varepsilon' n^2$; thus at least $2\varepsilon n^2$ changes are necessary to remove all copies of T_{F} in $T(\cdot, \cdot, z_1), T(\cdot, \cdot, z_1 + \frac{n}{2})$. There are $\frac{n}{2}$ disjoint pairs of such layers, so at least εn^3 changes are needed in total. T is therefore ε-far from being T_{F}-free. □

Observation 5.4.3. The property of 3-graphs of being free of the forbidden sub-3-graph, whose adjacency tensor is T_{F}, of the proof of Lemma 5.4.2, is hereditary — like any property of being free of forbidden induced substructures. It is also inflatable, as T_{F} is not a blowup of a smaller tensor, so F is not a blowup of a smaller 3-graph.

Before proceeding to the proof, we note that 3-graphs can be expressed as class of a general dense structures (as per Subsection 2.1.1): 3-partite structures, with a single ternary edge relation, constrained to only have edges with the first vertex of the tuple in the first vertex part, the second in the second part and the the third in the third vertex part. As the 3-graphs also have uniform order (the same number of vertices in each part), this implies that the results in Section 3.9 apply for 3-graphs.

Proof of Theorem 5.2. The proof is virtually the same as for Theorem 5.1.

Consider an ε-test of 3-graphs, making at most $q(\varepsilon)$ queries, for being free of the forbidden 3-hypergraph F whose adjacency tensor is T_{F} from Lemma 5.4.2. As by

Observation 5.4.3 this property is hereditary and inflatable, we may apply Corollary 3.9 to this test, which is specifically a uniform-order test, and obtain a perfectly canonical one-sided test for F-freeness with queried subgraph order $s(\varepsilon) = \mathrm{poly}(q(\varepsilon))$.

By Lemma 5.4.2, there exists a 3-dimensional tensor T of order n that is ε-far from being free of T_{F}, but only a $(c'/\varepsilon)^{c' \cdot \ln(c'/\varepsilon)}$ fraction of its order-2 subtensors are copies of T_{F}, for some global constant c'; let H be a 3-graph whose adjacency tensor is T.

The expected number of copies of F in a uniformly sampled sub-hypergraph of H is no more than $\mathrm{O}(s^6) \Big/ (c'/\varepsilon)^{c' \cdot \ln(c'/\varepsilon)}$ – the expected number of copies of T_{F} in a uniformly sampled subtensor of T of order s. Thus if $q(\varepsilon) < (c/\varepsilon)^{c \cdot \ln(c/\varepsilon)}$, for an appropriate constant c, then the expected number of copies of F discovered is $o(1)$, so the test accepts H with probability $1 - o(1)$ — a contradiction. □

Chapter 6

Pseudo-testing hypergraph tuple partition properties

6.1 Introduction

In this chapter we seek to further chart the territory of efficiently-testable properties of dense structures — specifically, hypergraphs with multiple edge relations or colors. For the case of dense graphs, [GGR98] established several specific properties to be testable using poly$(1/\varepsilon)$ queries — bipartiteness and colorability, max-clique, bisection, etc. — concluding with graph partition properties which can express all of these. Such properties are defined using a partition of the vertices (not the edges), with constraints on the sizes of the partition cells, and on the density of the bipartite graphs between edges (see [GGR98, Section 9] for the details). [GGR98] establishes their polynomial testability (albeit with running time exponential in the number of queries and $1/\varepsilon$). To date, this is the widest known "naturally-arising" class of polynomially-testable properties of dense graphs.

In [FMS07], Fischer, Matsliah and Shapira extended the polynomial testability of partition properties to hypergraphs with multiple edge relations. The constraints in this setting are very similar to the graph case: Constraints on the densities of vertex partition cells, and on the densities of the uniform hypergraphs with vertices originating in combinations of these partition cells.

A noteworthy use of this extension is its application to testing regular partitions in graphs (rather than hypergraphs): [GGR98] partition properties are not rich enough to express the constraint on a bipartite graph between two vertex sets of being regular in the sense of Szemerédi 's regularity lemma. Using hypergraphs, this constraint becomes expressible: Using the terminology of Gowers in [Gow07], a bipartite graph is regular if it has few 'combinatorial octahedra' — quadruples of vertices, two from each set, supporting a length-4 cycle. (This alternative view appears implicitly already in Alon, Duke, Lefmann, Rödl and Yuster's [ADL$^+$94].) One can construct an auxiliary hypergraph for a given graph, with an appropriate quaternary relation, and constrain it

to have few such octahedra; thus one can test, with a number of queries polynomial in ε, whether or not a graph has a regular partition with at most m cells. But while the broadening of the scope of testing to hypergraphs allowed for regular partitions of graphs, this kind of partition properties is not rich enough to express the constraints necessary to test for regular partitions of hypergraphs.

The motivation for considering a generalization of [GGR98] and [FMS07] partition properties is therefore double: The question of whether one can expand further the class of efficiently-testable properties (in hypergraphs, but possibly with implications on other structures); and the prospects for testing hypergraphs for regular partitions of a fixed size.

The generalization we make is that of partitioning not just the vertices of a hypergraph, but also tuples of multiple vertices — one partition of the singletons, another of the pairs, another of the triples, etc. Of course, this is meaningless unless the constraints on edges regard these partitions of tuples, so let us illustrate what this entails. All (hyper)edge constraints have the form "the density of k-vertex tuples, being edges of a certain color which satisfy some condition relating to the partition(s), out of the total n^s such tuples, is such-and-such". In [GGR98], the constraints are on 2-tuples (or on 2-sets, depending on whether the graph is directed), and the condition is "one vertex is in partition cell j_1, and the other vertex is in partition cell j_2". In [FMS07], the constraint is on any one of the hypergraph's edge relations, with its appropriate arity, but the condition is the same: The tuple is broken up into its s constituent vertices, and the origin of every one of them in the partition is constrained. Conditions regarding partitions of tuples will not always decompose a k-tuple into k single vertices, but rather make any sort of decomposition into tuples of arity up to k — designating certain pairs, triplets etc. of the elements of the tuple, with the condition being that each sub-tuple in the decomposition comes from some specified cell in the partition of tuple in its arity. Thus, taking 3-tuples for example, the condition may be that the pair of the first and third vertices come from cell no. 4 in the partition of pairs, while the second vertex of the 3-tuple comes from cell no. 5 of the partition of vertices.

While this generalization is not the broadest possible, it is the focus of this chapter, and it is already rich enough so that the results of [GGR98] and [FMS07] do not fully apply.

A key point in both these previous works is the following: If a (hyper)graph has a partition which approximately satisfies the density constraints, then the hypergraph is close to having a partition satisfying them exactly; that is, one can add or remove a small fraction of the edges so that a perfectly-satisfying partition is obtained. (In [FMS07], this point is made immediately after the statement of Theorem 2.) We show that this is not the case for partitions of hypergraph vertex tuples — at least not with a polynomial relation between the distance of the hypergraph from being satisfactory to the differences in densities of its best partition.

This difficulty is a corollary of two results, an upper bound and a lower bound, on testing such properties:

The upper bound result, in Section 6.3, is that one can 'test' whether a graph has a satisfying partition or whether all partitions are far from satisfying the constraints. This is not an actual test: While we accept graphs satisfying the property, we reject graphs not on account of their being far from satisfying it, but rather on account of their *partitions* being far from satisfactory. This 'pseudo-testing' can be done with a number of queries polynomial in $1/\varepsilon$, using a generalization of a technique from [GGR98] and [FMS07]: Beginning with an unknown satisfying partition, one can repeatedly redistribute more and more small sets of vertices and tuples, using 'type estimators' which minimize the 'damage' of this redistribution, so that the unknown partition which satisfies the constraints perfectly is gradually transformed into a partition which only approximately satisfies them, but is wholly known to the test. The validity of this final partition can be ensured with high probability of success. As pointed out above, for vertex-partition properties, this constitutes an actual test, but not so for tuple-partition properties.

The lower bound result, in Section 6.4, shows that some tuple partition properties are not polynomially testable. We demonstrate how tuple partition constraints are actually rich enough, already as we study them, to express the property of a graph being triangle-free. This gives a super-polynomial lower bound for testing tuple partition properties — at least in the general case. This lower bound does not rely on any specific triangle-testing lower bound construction (unlike the result in Chapter 5, which relies on a lower bound in partite graphs). We cannot even say for a fact that these partition properties are testable at all (that is, have tests independent of the size of the graph).

As mentioned above, the generalization in this work of partition properties is only partial. The expressive power it lacks is that of involving vertices and sub-tuples of constrained tuples in multiple conditions regarding the tuple partitions — cross-constraining, so to speak. For example, given a tuple $x = (x_1, \ldots, x_5)$, we might wish to constrain both the origin of (x_2, x_3, x_4, x_1) and at the same time also the origin of, say, (x_5, x_3, x_4). Such constraints are necessary for expressing hypergraph regularity, as the regular sub-entities of a hypergraph are simplical complexes, and their regularity involves densities of tuples supported by intersecting lower-arity tuples; for details, see one of the variant definitions of hypergraph regularity: [Gow07, NRS06, Ish09].

Attempts to establish the pseudo-testability of such properties have not met with success thus far. Some further discussion of the prospects for these more expressive partition properties is found in Chapter 7.

6.2 Additional preliminaries

6.2.1 Hypergraph tuple partition functions and named tuple decompositions

Definition 6.2.1. For some vertex set X and maximum arity k, *partition functions* with respect to X are in fact a single function over the domain $[k]$, but such that each function $\mathcal{P}(s)$ is a partition of the tuples of a certain arity into m sets, or cells: $\mathcal{P}(s) : \prod_{i=1}^{s} X \to [m]$.

We denote the j^{th} partition cell of arity s, with respect to partition functions $\mathcal{P}$, by $X_j^{\mathcal{P},s} = (\mathcal{P}(s))^{-1}(j)$.

Such partitions of the sets of tuples induce partitions of a hypergraph's set of *edges*, through the concept of edge decompositions which we define below.

Let $[s_1]^{\leq s_2}$ denote the set of all non-empty sequences, with length up to s_2, of distinct elements of $[s_1]$. For a sequence $A \in [s_1]^{\leq s_2}$, we denote by $\bigcup A$ the (unordered) set of all elements in A and by $|A|$ the length of A. Thus $A = \big(A_1, \ldots, A_{|A|}\big)$.

Definition 6.2.2. Let $x = (x_1, \ldots, x_s) \in \prod_{i=1}^{s} X$ and $A \in [s]^{\leq s}$. For tuple x and index sequence A, the *A-projection of x*, denoted $x(A)$, is the tuple y, of arity $|A|$, such that $y_i = x_{A_i}$.

Definition 6.2.3. A function ϕ with domain $\mathrm{Dom}(\phi) \subseteq [k']^{\leq k'}$ and range $[m]$ constitutes an *k'-named tuple decomposition* (or k'-NTD for short) if every pair of its constituent sequences $A, A' \in \mathrm{Dom}(\phi)$ are disjoint, i.e. $\bigcup A \cap \bigcup A' = \varnothing$, while on the other hand, all possible elements appear within some sequence in ϕ's domain: $\bigcup \{\bigcup A \mid A \in \mathrm{Dom}(\phi)\} = [k']$.
In other words, an s-NTD constitutes: a partition of $[k']$ (the coordinates in an k'-tuple); an ordering of the cells in this partition of the coordinates into sequences; and an indication for each such sequence of its intended origin in $\mathcal{P}$.

A tuple x is said to *observe* an NTD ϕ (by partition functions $\mathcal{P}$) if for every $A \in \mathrm{Dom}(\phi)$, $(\mathcal{P}(|A|))(x(A)) = \phi(A)$, i.e. the partition cell of the subtuple $x(A)$ of x is the one indicated by ϕ for A.

Example 6.2.4. Let $n = 2000$, $s = 3$, let $m = 10$ and let $x = (7, 1003, 21)$. Let ϕ be the NTD with domain $\mathrm{Dom}(\phi) = \{(2), (1, 3)\}$, so that $(2) \overset{\phi}{\mapsto} 5$ and $(1, 3) \overset{\phi}{\mapsto} 4$. The NTD ϕ represents the constraint on 3-tuples of their second element coming from the fifth partition cell (in a partition of individual vertices) and the subtuple of a 3-tuple, made up of its first and third element, coming from the fourth partition cell (of the partition of 2-tuples, which may be entirely unrelated to the partition of individual vertices). For x to observe ϕ with some partition functions $\mathcal{P}$, these must satisfy $(\mathcal{P}(1))(1003) = 5$ and $(\mathcal{P}(2))(1, 3) = 4$.

Note. NTDs will be used to impose constraints on vertex tuples, and offer a certain power of expressing such constraints. As discussed earlier, their expressive power is not maximal: No sub-tuple $x(A)$ of a tuple x has elements involved in two constraints of the same NTD simultaneously. In the above example, since the second element of the tuple is constrained as a singleton, no constraints on pairs can involve it. Thus instead of having $\mathrm{Dom}(\phi) = \{(1), (2), (3), (1,2), (2,3), (1,3)\}$ we only have the sequence of tuple indices $[s]$ decomposed into disjoint subsequences, each with its own constraint.

We denote by Φ^s the set of all s-NTDs, and their union of all arities up to k by $\Phi^{[k]} = \bigcup_{s \leq k} \Phi^s$.

6.2.2 Partitions and partition oracles

While we are interested in partitions of (multi-colored) hypergraphs, the objects we are testing are the hypergraphs themselves, rather than possible partitions; we will be constructing 'partition oracles' using queries to classify vertex tuples, thus simulating these hypothetical partitions.

Definition 6.2.5. A (q, m, k) *partition oracle* is a mapping $\pi : \bigcup_{s=1}^{k} \prod_{i=1}^{s} X \to [m]$ such that for $x \in \prod_{i=1}^{s} X$, $\pi(x)$ may be computed using q queries of the hypergraph. A *partial partition oracle* is defined similarly, but provides answers only for some subset $Y \subseteq \bigcup_{s=1}^{k} \prod_{i=1}^{s} X$.

Definition 6.2.6. A set of (possibly partial) oracles, sharing the same domain, is said to have *shared query complexity* q if, for any element of their domain, the set of all queries necessary for all the oracles to return an answer is of size at most q (where each of the oracles might use as many as all q queries).

Definition 6.2.7. Fix $\mathcal{P}(s)$, let $Y \subseteq \prod_{i=1}^{s} X$ and let $\mathcal{Q}(s)$ be a partition function for this subset. The *modification of* $\mathcal{P}(s)$ *according to* $\mathcal{Q}(s)$, is the function

$$(\mathcal{P}(s) \circ \mathcal{Q}(s))(x) = \begin{cases} (\mathcal{Q}(s))(x) & x \in Y \\ (\mathcal{P}(s))(x) & x \notin Y \end{cases}$$

Definition 6.2.8. For a partial partition oracle π for some set, the *modification of* $\mathcal{P}(s)$ *according to* π, denoted by $\mathcal{P}(s) \circ \pi$, is the modification of $\mathcal{P}(s)$ according to the partition induced by π.

6.2.3 Multi-colored hypergraph partition properties

Partition density features and density characteristics

As in the case of graph partition properties (studied in [GGR98]) and hypergraph vertex partition properties (studied in [FMS07]), we concern ourselves with the intersection of

the edge set(s) with sets of tuples obeying partition-related constraints. In the case of graphs, a constraint on an edge $\{x_1, x_2\}$ is "x_1 is in some certain specified cell of the vertex partition, and x_2 is in some certain specified cell"; for our partition properties, constraints correspond to NTDs.

Definition 6.2.9. For a hypergraph $\mathcal{H}$, partition functions $\mathcal{P}$ (with maximum arity $k \geq \max\{r(c) \mid c \in \mathcal{C}(\mathcal{H})\}$), a color $c \in \mathcal{C}(\mathcal{H})$ and an NTD $\phi \in \Phi^{r(c)}$, let

$$H_\phi(c) = \left\{ y \in H(c) \;\middle|\; \forall\, B \in \mathrm{Dom}(\phi)\ \left[\mathcal{P}(|B|)(y(B)) = \phi(B)\right] \right\}$$

that is, $H_\phi(c)$ is the set of all hyperedges in $H(c)$ which observe the NTD ϕ: For a subtuple of $[s]$ which ϕ constrains to some partition cell, $\mathcal{P}$ assigns the corresponding subtuple of y to the same partition cell.

Definition 6.2.10. An $(m, k, \mathcal{C})$-*density characteristic* is a tuple $\tau = (\rho, \mu)$ of density functions

$$\begin{aligned} \rho &: [k] \times [m] \to [0,1] \\ \mu &: \left\{ (c,\phi) \;\middle|\; c \in \mathcal{C} \text{ and } \phi \in \Phi^{r(c)} \right\} \to [0,1] \end{aligned}$$

ρ values shall be referred to as the characteristic's *tuple densities*, and μ values as its *edge densities*.

Definition 6.2.11. The *density characteristic corresponding to a hypergraph $\mathcal{H}$ and partition functions $\mathcal{P}$*, denoted $\psi^{\mathcal{H},\mathcal{P}} = \left(\rho^{\mathcal{H},\mathcal{P}}, \mu^{\mathcal{H},\mathcal{P}}\right)$, is the one satisfying, for each $k' \in [k]$ and $j \in [m]$,

$$\rho^{\mathcal{H},\mathcal{P}}(k', j) = \frac{1}{n^{k'}} \left| X_j^{\mathcal{P},k'} \right|$$

and for each $c \in \mathcal{C}(\mathcal{H})$ and $\phi \in \Phi^{r(c)}$,

$$\mu^{\mathcal{H},\mathcal{P}}(c, \phi) = \frac{1}{n^{r(c)}} |H_\phi(c)|$$

In other words, $\rho^{\mathcal{H},\mathcal{P}}(k', j)$ denotes the density of the j^{th} partition cell of k'-tuples within the entire set of k'-tuples; and $\mu^{\mathcal{H},\mathcal{P}}(c, \phi)$ denotes the density of the hypergraph's c-colored edges with the named decomposition ϕ.

Note. The edge density figures are 'absolute' — fractions of $n^{k'}$ possible tuples for some k', rather than fractions of the number of tuples with the same NTD.

Observation 6.2.12. The total number N^{dc} of vertex and edge density values in a single density characteristic is less than $k \cdot \left(m + c \cdot k^k \cdot m^k\right)$.

A hypergraph $\mathcal{H}$ and partition function $\mathcal{P}$ are said to *satisfy* a density characteristic ψ if $\psi^{\mathcal{H},\mathcal{P}} = \psi$. A hypergraph $\mathcal{H}$ is said to *satisfy* a density characteristic ψ if there exist partition functions which, together with $\mathcal{H}$, satisfy ψ.

Observation 6.2.13. A density characteristic ψ is satisfiable by hypergraphs on n vertices only if all tuple and edge densities of ψ are multiples of $n^{-k'}$ for the respective values of k'.

Definition 6.2.14. The *distance* between two density characteristics is the maximum difference between corresponding $\rho^{\mathcal{P}}(k', j)$ and $\mu^{\mathcal{P}}(c, \phi)$ values of the two characteristics.

Partition-based properties

Let Ψ be a set of $(m, k, \mathcal{C})$-density characteristics, and let $\mathcal{H}$ be a hypergraph on vertex set X with color set $\mathcal{C}$. $\mathcal{H}$ is said to *satisfy* the set Ψ if it satisfies some specific density characteristic $\psi \in \Psi$ (with some partition functions $\mathcal{P}$). $\mathcal{H}$ is said to *ε-approximately satisfy* Ψ if there exist partition functions $\mathcal{P}$, and some $\psi \in \Psi$, which is in itself satisfiable and of distance at most ε from $\psi^{\mathcal{H},\mathcal{P}}$.

Definition 6.2.15. For a density characteristic set Ψ, the *property* Π_Ψ of hypergraphs is defined as the set of all hypergraphs which satisfy Ψ.

A hypergraph is said to *ε-approximately satisfy* Π_Ψ if it ε-approximately satisfies Ψ. As discussed in the introduction to this section, a hypergraph's being approximately satisfying of Π_Ψ does not necessarily imply that it is also close to satisfying Π_Ψ. We thus make a definition analogous to that of a property test.

Definition 6.2.16. Let Π_Ψ be a partition property as per the above. A *pseudo-test* for Π_Ψ is a probabilistic oracle machine with the same input and oracle as a (dense model) property test, which distinguishes with probability at least $2/3$ between the case of G being in Π_Ψ and the case in which, for every choice of partition functions $\mathcal{P}$, $\left(\rho^{\mathcal{H},\mathcal{P}}, \mu^{\mathcal{H},\mathcal{P}}\right)$ is ε-far from Ψ (rather than the case of G being far from Π_Ψ).

Pseudo-testing can be seen as testing under a different distance metric — the minimum distance of the hypergraphs' partition functions.

6.2.4 Tuple types and type estimators

What is the effect of reassigning a hypergraph's vertex tuple a different partition cell on the partition's density characteristic? As in [GGR98] and [FMS07], we will need to estimate this effect and cluster tuples accordingly, so as to be able to redistribute tuples among partition cells without affecting the partition density characteristic overmuch.

Let $s \leq k' \leq k$, let $x \in \prod_{i=1}^{s} X$ and let A be a sequence in $[k']^s$ (that is, a sequence of s distinct elements between 1 and k'). We say that *x takes the role of A* in a tuple $y \in \prod_{i=1}^{k'} X$ if $y(A) = x$. We're interested in characterizing the effect on edge densities of having x taking the role of different sequences A, for every possible decomposition of the rest of the tuple besides x:

Definition 6.2.17. A partial function $\phi : [k']^{\leq k'-s} \to [m]$ constitutes an *A-less k'-named tuple decomposition for arity k'* (or (k', A)-NTD for short) if for every two

distinct sets $B, B' \in \mathrm{Dom}(\phi)$, $\bigcup B \cap \bigcup B' = \varnothing$, and every index not in A is covered by some B, while no index in A is covered, i.e. $\bigcup\{\bigcup B \mid B \in \mathrm{Dom}(\phi)\} = [k'] \setminus \bigcup A$.

The set of all (k', A)-NTDs shall be denoted $\Phi^{k',A}$, with $\Phi^{[k],A}$ denoting their union over all $k' \in [k]$.

Definition 6.2.18. For a hypergraph $\mathcal{H}$, partition functions $\mathcal{P}$, a color $c \in \mathcal{C}(\mathcal{H})$, an $(r(c), A)$-NTD $\phi \in \Phi^{r(c),A}$, and an $|A|$-tuple x, we define

$$H_\phi^{A,x}(c) = \left\{ y \in H(c) \;\middle|\; y(A) = x \text{ and } \forall\, B \in \mathrm{Dom}(\phi)\ \left[\mathcal{P}(|B|)(y(B)) = \phi(B)\right] \right\}$$

In other words,$H_\phi^{A,x}(c)$ is the set of all hyperedges in $\mathcal{H}$ of color c, in which x takes the role of A, and which also observe ϕ.

Definition 6.2.19. An *s-tuple type* is a function $\tau : \mathrm{TypeDom}\,(s) \to [0,1]$, with its domain being

$$\mathrm{TypeDom}\,(s) = \bigcup_{k'=s}^{k} \left\{ (A, c, \phi) \in [k]^s \times \mathcal{C}(\mathcal{H}) \times \Phi^{[s],A} \;\middle|\; r(c) = k' \text{ and } \phi \in \Phi^{r(c),A} \right\}$$

Definition 6.2.20. The *type of an s-tuple x* with respect to a hypergraph $\mathcal{H}$ and partition functions $\mathcal{P}$ is the s-tuple type $\tau^{\mathcal{H},\mathcal{P},x}$, whose values are the relative sizes of all of constrained edge sets of the various arities and for the various roles x can take in such edge sets:

$$\tau^{\mathcal{H},\mathcal{P},x}(A, c, \phi) = \frac{1}{n^{(r(c)-s)}} \left| H_\phi^{A,x}(c) \right|$$

We denote by $\mathsf{Type}(s)$ the set of all possible types of s-tuples (with respect to m and k).

Definition 6.2.21. The *distance* $\mathrm{dist}\big(\tau_1, \tau_2\big)$ between two s-tuple types is the maximum over $(A, c, \phi) \in \mathrm{TypeDom}\,(s)$ of the absolute differences $|\tau_1(A, c, \phi) - \tau_2(A, c, \phi)|$.

In our arguments below we shall be using rounded estimates of tuples' type for clustering. For this purpose, we define:

Definition 6.2.22. The *tuple type ε-net for s-tuples* is the set $\mathrm{TypeNet}_{s,\varepsilon}$ of all types $\tau = (\rho, \mu)$ supported on exact multiples of ε.

Lemma 6.2.23. *The size of the ε-net for s-tuples is polynomial in $1/\varepsilon$:*

$$\left|\mathrm{TypeNet}_{s,\varepsilon}\right| < (1 + 1/\varepsilon)^{|\mathcal{C}(\mathcal{H})| \cdot m^{(2ek)^{2k}}} = \mathrm{poly}(1/\varepsilon)$$

Proof. For every one of the $|\mathcal{C}(\mathcal{H})|$ colors, $\mathrm{TypeDom}\,(s)$ has elements for any possible choice of a sequence A of length s and an A-less NTD in $\Phi^{r(c)}$. The number of such choices is m to the power of possible decompositions of an $r(c)$-tuple into a sequence A and additional subsequences of $[r(c)] \setminus A$. The number of such decompositions can be upper-bounded as follows: Order the elements of $[r(c)]$; the first s elements constitute

A; as for the rest, one has to choose sizes for as many as $r(c) - s$ additional sequences (which contain the remaining elements of the $r(c)$-tuple as ordered); assuming $s < r(c)$, the number of such choices is $\binom{2(r(c)-s)-1}{r(c)-s}$. The number of decompositions is therefore under $(r(c))! \cdot \binom{2(r(c)-s)-1}{r(c)-s} < (2er(c))^{2r(c)} < (2ek)^{2k}$. The claim follows. □

$\text{TypeNet}_{s,\varepsilon}$ is a $\frac{1}{2}\varepsilon$-net of tuple types — any type is within a distance of $\frac{1}{2}\varepsilon$ of a type in $\text{TypeNet}_{s,\varepsilon}$. It can therefore induce a clustering of types, associating each possible type with one of those in $\text{TypeNet}_{s,\varepsilon}$: We first impose some arbitrary order on $\text{TypeNet}_{s,\varepsilon}$, then define:

Definition 6.2.24. For any s-tuple type τ, the $\text{TypeNet}_{s,\varepsilon}$ *type corresponding to* τ is the first type among those $\text{TypeNet}_{s,\varepsilon}$ types which is at the minimum distance from τ.

Abusing notation, we refer to this corresponding type in $\text{TypeNet}_{s,\varepsilon}$ as "τ's cluster" with respect to $\text{TypeNet}_{s,\varepsilon}$. Given a specific hypergraph and partition functions $\mathcal{P}(\cdot)$, this clustering of tuple types also induces a clustering of a hypergraph's tuples — a clustering according to type.

Aside from the single type with which a tuple x is associated, x is said to be *compatible* with any tuple type $\tau \in \text{TypeNet}_{s,\varepsilon}$ whose distance from τ is less than ε.

Definition 6.2.25. A *type estimator* for a set $Y \in \prod_{i=1}^{s} X$ with respect to $\text{TypeNet}_{s,\varepsilon}$ is a probabilistic machine which, given some tuple $y \in Y$ as input, makes certain queries to the hypergraph, and then returns an element of $\text{TypeNet}_{s,\varepsilon}$ as the estimated cluster of y.

The concept of *shared query complexity* for type estimators is defined similarly to the case of partition oracles (see Definition 6.2.6).

6.3 An upper bound on pseudo-testing partition properties

We begin with our positive result regarding tuple-partition properties: The possibility of efficiently distinguishing hypergraphs with satisfying partitions from hypergraphs which do not ε-approximately satisfy the given constraints with any partition:

Theorem 6.1. *Let Ψ be a set of density characteristics for hypergraphs with colors $\mathcal{C}$, regarding partitions with m cells in each arity. One can pseudo-test Π_Ψ with a number of queries polynomial in ε.*

Note. For the purpose of this theorem, we assume that the set Ψ is 'easy', in the sense that one can efficiently compute the distance of a specific density characteristic from Ψ (and specifically, whether it is in Ψ or not). We omit an exact definition of this notion.

The key to the proof (similarly to the argument in [GGR98, Section 9.1] and [FMS07, Sections 6,7]) is the following: One can arbitrarily redistribute small sets of tuples among the m partition cells (of the appropriate arity in our case), with a relatively small effect on the partition's density characteristic — provided that the elements in the small set being redistributed all have very similar types, and that the overall size of every partition cell remains almost the same after redistribution. This is established in Subsection 6.3.1. Given a small set $Y \subseteq \prod_{i=1}^{s} X$, and assuming that the rest of the partition is known to us, we can rebuild another, similar, partition resulting from the small-set redistribution. We can do so repeatedly for a chosen partition of all the vectors in $\prod_{i=1}^{1} X, \ldots, \prod_{i=1}^{k} X$ into such small redistribution sets, so that, in fact, we eventually need not have any output depend on knowledge of the original partition; the overall deviation from the original partition's density characteristic will still be relatively small.

The problem with this procedure is that for every small set Y we examine, we do not actually know the rest of the partition, nor the fraction of the elements in Y of each type within each of the partition cells. We overcome this ignorance by simply trying all possibilities, i.e. when sampling tuples with which to determine the type of the elements of Y, we will 'branch' our computation for all m partition cells to which any tuple may belong. Similarly, when choosing how to redistribute the elements of Y of similar type, we will in fact branch our computation for all possible sizes for distribution among the m partition cells (rounded to multiples of some fraction depending on ε). We will thus construct, in fact, a large number of partition oracles — exponential in $1/\varepsilon$ — but their shared query complexity will still be polynomial in $1/\varepsilon$, as they all use the same set of queries. This construction of partition oracles and tuple type estimators is described and analyzed in Subsection 6.3.2.

If an appropriate partition exists, then one of these oracles will simulate it relatively well. The pseudo-test will be able to determine whether this is indeed the case by estimating the partition's density characteristic using the oracle. This is demonstrated in Subsection 6.3.3, and allows us to complete the proof.

Throughout this section we assume that $k \geq \max\{r(c) \mid c \in \mathcal{C}(\mathcal{H})\}$, and without loss of generality that $k = \max\{r(c) \mid c \in \mathcal{C}(\mathcal{H})\}$. Also, our query complexity expressions treat m and k as constants rather than parameters (e.g. the $O(\cdot)$ notations hide coefficients depending only on m and k).

6.3.1 Key Lemma: Low-damage tuple redistribution

Our proof hinges on repeatedly estimating the types of vertex tuples — with respect to initial partitions $\mathcal{P}$ — and then modifying $\mathcal{P}$ by redistributing tuples of similar type among the various cells at a given arity, while ensuring that $\psi^{\mathcal{H},\mathcal{P}}$ does not change overmuch.

Suppose, then, that we have a small set of tuples to redistribute. Formally, let $\tau \in \mathsf{Type}(s)$ be a type of s-tuples. Let $Y^{\tau} \subseteq \prod_{i=1}^{s} X$ be a small set of tuples with

$|Y^\tau| < \zeta n^s$, whose types are all ε'-compatible with τ with respect to the partition $\mathcal{P}(s)$, except perhaps for at most a ξ-fraction. Also, let $\mathcal{Q} : Y^\tau \to [m]$ be a re-distribution of Y^τ — a function partitioning it into m cells, which maintains fairly well the number of elements in each partition cell: The partition cell sizes of $\mathcal{Q}$ and $\mathcal{P}(s)$ (with respect to Y^τ) differ by at most an η-fraction of the total size, that is, for every $j \in [m]$,

$$\left|\mathcal{Q}^{-1}(j) - \left|X_j^{\mathcal{P},s} \cap Y^\tau\right|\right| < \eta|Y^\tau|$$

Lemma 6.3.1. *Let $\mathcal{P}$, Y^τ, $\mathcal{Q}$ be as per the above, and denote $\tilde{\mathcal{P}} = \mathcal{P} \circ \mathcal{Q}$. Then*

$$\mathrm{dist}\big(\psi^{\mathcal{H},\mathcal{P}}, \psi^{\mathcal{H},\tilde{\mathcal{P}}}\big) < k^2\big(\varepsilon' + \eta + \zeta + \xi\big) \cdot \frac{|Y^\tau|}{n^s}$$

Proof. We must bound the change in density of both the vertex and the edge densities of $\psi^{\mathcal{H},\tilde{\mathcal{P}}}$ relative to $\psi^{\mathcal{H},\mathcal{P}}$.

As regards the tuple densities of $\tilde{\mathcal{P}}$, the claim follows from the fact that each set $X_j^{\tilde{\mathcal{P}},s}$ has lost $\left|X_j^{\mathcal{P},s} \cap Y^\tau\right|$ elements and gained $\left|\mathcal{Q}^{-1}(j)\right|$ elements; the density $\rho(s,j)$ has therefore changed by no more than $\eta \cdot n^{-s}|Y^\tau|$ as per the constraint on $\mathcal{Q}$.

Moving to edge densities, fix some color $c \in \mathcal{C}$ and an NTD $\phi \in \Phi^{r(c)}$; we must bound the change in the density $H_\phi(c)$. We do so by considering various kinds of s-tuples in $H_\phi(c)$ with respect to before and after the redistribution:

First consider those $r(c)$-tuples x containing at least two s-subtuples from Y^τ: $x(A) \subseteq Y^\tau$ and $x(B) \in Y^\tau$ for two different (and disjoint) sequences $A, B \in \mathrm{Dom}(\phi)$. The fraction of these tuples within all $r(c)$-tuples is at most $n^{-2s}|Y^\tau|^2 < \zeta \cdot n^{-s}|Y^\tau|$ for every choice of disjoint sequences A and B in ϕ; the number of such choices is less than $(r(c)/s)^2 \leq r(c)^2$, so the total contribution of such tuples to the change in density is less than $r(c)^2 \cdot \zeta \cdot n^{-s}|Y^\tau|$.

Next, consider some maximal 1:1 relation between $\mathcal{Q}^{-1}(\{j\})$ and $X_j^{\mathcal{P},s} \cap Y^\tau$; we can think of the sources of this relation as tuples being replaced by tuples of similar type (with the rest of the tuples being removed-only or added-only). Consider such a pair of s-tuples, y and y', and assume that both have a type which is ε'-compatible with τ. This replacement effects $H_\phi(c)$ through $r(c)$-tuples in which either y or y' appears at least once. Consider some $A \in \mathrm{Dom}(\phi)$ of length s, and let $\phi' \in \Phi^{r(c),A}$ be the corresponding $(r(c),A)$-NTD (obtained by removing $A \mapsto j$ from ϕ). We have $\left|\tau^{\mathcal{P},y}(A,c,\phi') - \tau^{\mathcal{P},y'}(A,c,\phi')\right| < \varepsilon'$, so $H_\phi(c)$ gains or loses at most $\varepsilon' n^{r(c)-s}$ tuples by replacing $H_{\phi'}^{A,y}(c)$ with $H_{\phi'}^{A,y'}(c)$. Summing over all possible choices of $A \in \mathrm{Dom}(\phi)$ (at most $r(c)/s \leq r(c)$ of these) and all pairs y, y' in the matching, we find that $H_\phi(c)$ gains or loses at most $r(c) \cdot \varepsilon' \cdot n^{r(c)-s} \cdot |Y^\tau|$, i.e. its density changes by at most $r(c) \cdot \varepsilon' \cdot n^{-s}|Y^\tau|$.

In this last estimate we have disregarded the effect of $r(c)$-tuples with more than one s-subtuple from Y^τ taking the place of some $A \in \phi$ — these may behave differently than what the type of an individual Y^τ s-tuple suggests, but the aggregate contribution of all such tuples to the change in density has already been accounted for with the bound

involving ζ above. We have also disregarded the effect of tuples with incompatible type, which will be considered below.

Now consider those s-tuples in $\mathcal{Q}^{-1}(\{j\})$ or $X_j^{\mathcal{P},s} \cap Y^\tau$ which are neither sources nor targets in the above-mentioned 1:1 relation. Their number is at most the difference in size between $\mathcal{Q}^{-1}(\{j\})$ and $X_j^{\mathcal{P},s} \cap Y^\tau$, which by our assumptions does not exceed $\eta|Y^\tau|$; thus the fraction of $r(c)$-tuples in which they take the role of some $A \in \mathrm{Dom}(\phi)$, and their effect on the density of $H_\phi(c)$, is at most $r(c) \cdot \eta \cdot n^{-s}|Y^\tau|$.

Finally, for every $A \in \mathrm{Dom}(\phi)$, there are at most $\xi \cdot |Y^\tau| \cdot n^{r(c)-s}$ $r(c)$-tuples in which the role of A is taken by a Y^τ element whose type is incompatible with τ. The total contribution of these tuples over all $A \in \mathrm{Dom}(\phi)$ to the change in $H_\phi(c)$ density is at most $r(c) \cdot \xi \cdot n^{-s}|Y^\tau|$.

All other $r(c)$-tuples do not involve s-tuples from Y^τ, are neither introduced into $H_\phi(c)$ nor removed from it by the redistribution of Y^τ, and do not affect changes in its density.

Summing up the above (and recalling that $r(c) \leq k$) yields the claim regarding the edge density $\mu(c, \phi)$, for any possible choice of c and ϕ. □

6.3.2 Generating type estimators and partition oracles

The first two of the following three lemmata each requires the next one in its proof; the proofs appear after the statement of all three. Note that the complexity bounds in all these lemmata treat k and m as constants rather than parameters.

Lemma 6.3.2. *Let Ψ be a set of density characteristics, and let $\delta, \varepsilon' > 0$. One may generate a set $S_{oracles}$ of $\exp(\mathrm{poly}(1/\varepsilon') \cdot \ln(1/\delta))$ partition oracles for $\mathcal{H}$ with shared query complexity $q_{6.3.2}(\varepsilon') = \mathrm{poly}(1/\varepsilon') \cdot O(\ln(1/\delta))$, such that if the hypergraph satisfies Ψ, then with probability at least $1 - \delta$ at least one of these oracles induces partition functions which $\frac{1}{2}\varepsilon'$-approximately satisfy Ψ. This, without making any queries to $\mathcal{H}$, and independently of Ψ.*

Lemma 6.3.3. *Let $\tilde{\mathcal{P}}$ be partition functions for a hypergraph $\mathcal{H}$, let $Y \subseteq \prod_{i=1}^{s} X$ be a set of normalized size at most $1/\ell$ and let $\delta' > 0$. One may generate a set $S_{oracles}^{Y}$ of $\exp(\mathrm{poly}(\ell)\ln(1/\delta'))$ partial oracles for s-tuples in Y, with shared query complexity $q_{6.3.3}(\ell) = \mathrm{poly}(\ell) \cdot O(\ln(1/\delta'))$, so that with probability at least $1 - \delta'$, at least one of these oracles (say, π) is such, that the partition functions $\tilde{\mathcal{P}} \circ \pi$ $\left(6k^2/\ell^2\right)$-approximately satisfy $\psi^{\mathcal{H},\tilde{\mathcal{P}}}$. This, without making any queries to the hypergraph and independently of $\tilde{\mathcal{P}}$.*

Lemma 6.3.4. *Let $\tilde{\mathcal{P}}$ be partition functions for a hypergraph $\mathcal{H}$, let $Y \subseteq \prod_{i=1}^{s} X$ and let $\delta'', \varepsilon'', \xi > 0$. One may generate a set of at most $\exp(\mathrm{poly}(1/\varepsilon'') \cdot \ln(1/\delta''\xi))$ type estimators for the tuples in Y, all using a single uniformly-sampled sequence U of $\mathrm{poly}(1/\varepsilon'') \cdot O(\ln(1/\delta''\xi))$ vertices, such that at least one of these oracles suggests a compatible cluster with respect to $\mathrm{TypeNet}_{s,\varepsilon''}$ and $\tilde{\mathcal{P}}$ for all but a ξ-fraction of the tuples*

in each of the clusters induced by $\mathrm{TypeNet}_{s,\varepsilon''}$*, and such that all oracles only query tuples involving vertices from the input tuple and from* U*, for a shared query complexity of* $\Theta(|U|)$*. This, with probability at least* $1-\delta''$ *over the choice of* U*, independently of* $\tilde{\mathcal{P}}$*, and with no queries made in advance so as to obtain the oracles.*

Proof of Lemma 6.3.2. Set $\ell = 12k^3/\varepsilon'$. Assume that $\mathcal{H}$ does indeed satisfy the density characteristic Ψ with partition functions $\mathcal{P}$. At every arity $s \leq k$, we choose an arbitrary partition of the s-tuples into ℓ equal-size sets: $\{Y_{s,1}, \ldots, Y_{s,\ell}\}$.

We generate small-set oracles as described in Lemma 6.3.3, with $\delta' = \delta/2k\ell$, obtaining sets S^Y_{oracles} of partial oracles for each subset at each arity.

We will now transition from the initial partition functions $\mathcal{P}$ through a sequence of intermediary partition functions, up to the final partition functions $\tilde{\mathcal{P}}^{k,\ell}$, which will still approximately-satisfy Ψ, even though their assignment of cells to tuples is based wholly on the partitions into $Y_{s,j}$ sets. At every subsequent transition, we apply Lemma 6.3.3 regarding one of the $Y_{s,j}$ sets, to obtain partial oracles for this set — but with respect to the previous intermediary partition functions, rather than with respect to the initial partitions $\mathcal{P}$. This is possible due to the fact that Lemma 6.3.3 applies regardless of the partition for which partial oracles are sought.

Indeed, set $\tilde{\mathcal{P}}^{0,\ell}$ to $\mathcal{P}$, satisfying Ψ exactly. With probability at least $1-\delta/2k\ell$, one of the oracles for $Y_{1,1}$, call it $\pi_{1,1}$, is such that $\tilde{\mathcal{P}}^{1,1}$ $6k^2/\ell^2$-approximately satisfies $\psi^{\tilde{\mathcal{P}}^{0,\ell}}$ (specifically, $\mathcal{P} \circ \pi_{1,1}$ $\varepsilon'/2k\ell$-approximately satisfies $\psi^{\mathcal{H},\mathcal{P}}$; and this will hold for subsequent partition functions at any arity, not just arity 1, by our choice of ℓ). Similarly, with probability at least $1-\delta/2k\ell$, one of the oracles for $Y_{1,2}$ is such that $\tilde{\mathcal{P}}^{1,2} = \tilde{\mathcal{P}}^{1,1} \circ \pi_{1,2}$ $\varepsilon'/2k\ell$-approximately satisfies $\psi^{\tilde{\mathcal{P}}^{1,1}}$ (thus $\varepsilon'/k\ell$-approximately satisfying $\psi^{\tilde{\mathcal{P}}^{0,\ell}}$), and so on until $\tilde{\mathcal{P}}^{1,\ell}$ which $\varepsilon'/2k$-approximately satisfies $\psi^{\tilde{\mathcal{P}}^{0,\ell}}$. We implicitly construct similar partition functions $\tilde{\mathcal{P}}^{s,1}, \ldots, \tilde{\mathcal{P}}^{s,\ell}$ for the sets of 2-tuples, 3-tuples, and every arity s — beginning each time with $\tilde{\mathcal{P}}^{s-1,\ell}$ from the previous phase. Eventually, with probability at least $1-\delta/2$, some sequence of oracles $(\pi_{1,1}, \ldots, \pi_{k,\ell})$ yields a complete partition $\tilde{\mathcal{P}}^{k,\ell}$ which $\frac{1}{2}\varepsilon'$-approximately satisfies $\tilde{\mathcal{P}}^{0,\ell} = \tilde{\mathcal{P}}$.

Consequently, our set of oracles for the entire hypergraph is the set of all combinations of $Y_{s,1}, \ldots, Y_{s,\ell}$ oracles for all $s \in [k]$, constituting $(\exp(\mathrm{poly}(\ell) \cdot \ln(1/\delta')))^{k\ell} = \exp(\mathrm{poly}(1/\varepsilon') \cdot \ln(1/\delta))$ oracles in total. Their shared query complexity is the same as for the small-set partial oracles — as in order to get the oracles' output for a given vertex tuple, one in fact uses only the small-set oracles 'covering' the tuple in question.□

Proof of Lemma 6.3.3. Our partition oracles will be based on the principle of 'low-damage' small-set redistribution, embodied in Lemma 6.3.1: We will estimate the types of the various tuples of Y, and redistribute them accordingly.

Assume initially that $\tilde{\mathcal{P}}$ is known to us; we will later forego this assumption.

First, we choose one of the type estimators of Lemma 6.3.4 — with parameters $\xi = 1/\ell$, $\delta'' = \delta'$ and $\varepsilon'' = 1/\ell$; the estimator induces a clustering of the s-tuples in Y by their estimated type, which we denote $\left\{Y^\tau \;\middle|\; \tau \in \mathrm{TypeNet}_{s,\varepsilon''}\right\}$. Our redistribution

will respect the lexicographic order of s-tuples, so that all tuples in Y up to some tuple x_1 are reassigned to cell 1, tuples between x_1 and x_2 are reassigned to cell 2 etc. This will later allow us to forego the knowledge of $\tilde{\mathcal{P}}$; for now note that the decision of how to redistribute tuples with a similar type does not have any affect on the applicability of Lemma 6.3.1.

We must this decide, for each Y^τ and each partition cell $j < m$, at which tuples to make the transition from cell j to cell $j+1$. This is clearly dictated by the size of the intersection of Y^τ with each cell j; but instead of using the exactly appropriate tuple range, we set the ranges differently: We only choose as boundary tuples for cell reassignment such tuples whose positions are multiples of $n^s/\ell^3\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$, between 0 and n^s. For each cell j the choice is either of the highest multiple of $n^s/\ell^3\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$ below $\left|X_j^{\tilde{\mathcal{P}},s} \cap Y^\tau\right|$ (the original intersection size), or the lowest multiple above $\left|X_j^{\tilde{\mathcal{P}},s} \cap Y^\tau\right|$. The decision of which of these options to choose is made so that the ranges cover all n^s tuples exactly. (This choice is possible, since we can begin by always choosing the lower multiple of $n^s/\ell^3\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$ for the differences in boundaries, ending up not covering all n^s tuples, and then gradually increasing the differences to the higher multiples; at some point we will hit n^s exactly).

We now redistribute Y^τ according to the boundaries we have set. If Y^τ is very small, it is possible that we've changed the intersection sizes by a significant fraction of the size of Y^τ — perhaps even placed all of it in a single partition cell. But for most Y^τ this is not the case: Denoting by $\mathrm{T}^{\mathrm{small}}$ the set of types $\tau \in \mathrm{TypeNet}_{s,\varepsilon''}$ for which $|Y^\tau| < n^s/\ell^2\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$, we have $\left|\bigcup\left\{Y^\tau \;\middle|\; \tau \in \mathrm{T}^{\mathrm{small}}\right\}\right| < n^s/\ell^2$, as $\mathrm{T}^{\mathrm{small}}$ has no more than $\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$ elements. For a $\tau \notin \mathrm{T}^{\mathrm{small}}$, the size of each of its intersections with each partition cell changes by at most $|Y^\tau|/\ell$ relative to the original partition.

This redistribution has in general an adverse effect on $\tilde{\mathcal{P}}$'s satisfaction of Ψ: Even if Y^τ is not very small, and if the types of all tuples in Y^τ were exactly τ, and the redistribution would not be changing the sizes of partition cells' intersections with Y^τ sets at all — there would still be the effect of tuples involving multiple elements from Y^τ which have now changed cells. And of course, the type estimators may not be perfectly exact; and the types in Y^τ are only close to τ; and the redistribution intersection sizes are only close to the original sizes. Still, we can apply Lemma 6.3.1, to bound the effect of the redistribution on the the density characteristic: For a Y^τ with $\tau \in \mathrm{T}^{\mathrm{small}}$, Lemma 6.3.1 applies with parameters $\zeta = 1/\ell^2\big|\mathrm{TypeNet}_{s,\varepsilon''}\big|$, $\xi = \varepsilon' = 1/\ell$ and $\eta \leq 1$; and for Y^τ with $\tau \notin \mathrm{T}^{\mathrm{small}}$, the lemma applies with $\zeta = \xi = \eta = \varepsilon' = 1/\ell$.

Let us sum up the total effect of these redistributions as a bound on the distance from the original partition (using the triangle inequality). The contribution of the redistribution of Y^τ with $\tau \in \mathrm{T}^{\mathrm{small}}$ is at most $k^2\big(1/\ell^2\big|\mathrm{TypeNet}_{s,\varepsilon''}\big| + 1/\ell + 1 + 1/\ell\big) \cdot n^{-s}|Y^\tau| < 2k^2 n^{-s}|Y^\tau|$; over all such sets Y^τ the total contribution is at most $2k^2 n^{-s} \cdot \big(n^s/\ell^2\big) = 2k^2/\ell^2$. The contribution of the redistribution of a Y^τ set with $\tau \notin \mathrm{T}^{\mathrm{small}}$ is at most $k^2(1/\ell + 1/\ell + 1/\ell + 1/\ell) \cdot n^{-s}|Y^\tau|$, and over all such Y^τ, at most $4k^2/\ell^2$.

Thus, if one of the type estimators of Lemma 6.3.4 clusters most vertices in each cluster into compatible clusters (which happens with probability at least $1 - \delta'$), then the choice of this type estimator yields a partition $\tilde{\mathcal{P}} \circ \pi$ (our initial partition following the redistribution of the Y^τs) which $6k^2/\ell^2$-approximately satisfies $\psi^{\mathcal{H},\tilde{\mathcal{P}}}$.

We need, however, to achieve the above without foreknowledge of $\tilde{\mathcal{P}}$. We note that the only use of the knowledge of $\tilde{\mathcal{P}}$ in the argument above was the choice of boundary values for the redistribution of each Y^τ, and even those were only multiples of $|Y|/\ell^2\left|\mathrm{TypeNet}_{s,\varepsilon}\right|$. Thus instead of relying on our knowledge of the partition, we will have multiple oracles, one for every possible setting of boundary values for Y^τ, for every type $\tau \in \mathrm{TypeNet}_{s,\varepsilon'}$, and every one of the m partition cells in $\tilde{\mathcal{P}}(s)$. (Note that each such oracle for Y can readily compute the redistribution cell for a given tuple using the type estimate and its predefined boundary values.) The total number of such configurations is less than $\left(\ell^3\left|\mathrm{TypeNet}_{s,\varepsilon}\right| + 1\right)^{m\cdot\left|\mathrm{TypeNet}_{s,\varepsilon'}\right|}$, so the total number of oracles for establishing the claim is this number, times the number of possibilities for a choice of the type estimator from Lemma 6.3.4. By Lemma 6.2.23, the first multiplicand is $O(\exp(\mathrm{poly}(\ell)))$; by Lemma 6.3.4, the second multiplicand is $p_{6.3.4}(1/\ell, \delta', 1/\ell) = \exp(\mathrm{poly}(\ell) \cdot \ln(\ell/\delta')) = \exp(\mathrm{poly}(\ell) \cdot \ln(1/\delta'))$, so the product is $\exp(\mathrm{poly}(\ell) \cdot \ln(1/\delta'))$ oracles overall, as claimed. With probability at least $1-\delta'$, at least one of the choices of the type estimator and one of the choices of rounded intersection values correspond well enough to the actual partition so that Lemma 6.3.1 applies with the above parameters. The oracles maintain the same shared query complexity as that of a single oracle, since they do not differ with respect to the queries made for a given tuple, so the same queries can be used by all oracles; this query complexity is, in turn, merely that of using the type estimator, i.e. $O(\ln(\ell/\delta')) \cdot \mathrm{poly}(\ell) = \mathrm{poly}(\ell) \cdot O(\ln(1/\delta'))$ as claimed. □

Proof of Lemma 6.3.4. Our type estimators will base their output on the clustering induced by $\mathrm{TypeNet}_{s,\varepsilon''}$ — applied to an estimate of a tuple's type rather than its actual type. We assume initially that the partition functions $\tilde{\mathcal{P}}$ are known, and describe a single oracle clustering the tuples.

Let $y \in Y$ be the tuple to have its type clustered. Set

$$t = \ln\left(\frac{1}{\delta''\xi} \cdot N^{\mathrm{dc}} \cdot \left|\mathrm{TypeNet}_{s,\varepsilon''}\right|\right) \cdot \frac{2}{\varepsilon''^2}$$

where N^{dc} is the number of density values in a density characteristic (see Observation 6.2.12).

Let $(A, c, \phi) \in \mathrm{TypeDom}\,(s)$. The estimator samples, uniformly, a sequence of $r(c) - s$ vertices from X, which complete y into an $r(c)$-tuple x with $x(A) = y$. Using its knowledge of the partition, the estimator determines whether or not this tuple observes the NTD ϕ, and queries the hypergraph to determine whether x is an edge of $H(c)$. This is repeated t times, independently, and the density value estimate $\tau^{U,y}(A, c, \phi)$ is

the fraction of samples x in $H_\phi(c)$.

This estimate is made for all $(A, c, \phi) \in \mathrm{TypeDom}\,(s)$. As we will be union-bounding the probability of any of the estimates deviating overmuch, we have the estimator use the same samples for all choices of (A, c, ϕ), that is, a sequence of $k - s$ vertices is sampled t times, with only the first $r(c) - s$ vertices in each sample are used for $\tau^{U,y}(A, c, \phi)$ estimates. This entire sequence of $t \cdot (k - s)$ vertices is our choice of U.

The estimation of a tuple's type may fail (with the estimator suggesting an incompatible cluster) only if one of the estimated density values is $\frac{1}{2}\varepsilon''$-far from the actual value. The probability that a uniformly sampled completion of x into an $r(c)$-tuple will be in $H_\phi(c)$ is, by definition, $\tau^{\tilde{\mathcal{P}},y}(A, c, \phi)$, and the estimate $\tau^{U,y}(A, c, \phi)$ is an average of t independent indicators with this probability. We may therefore apply a large deviation bound to conclude that

$$\mathbf{Pr}\left[\left|\tau^{U,y}(A,c,\phi) - \tau^{\tilde{\mathcal{P}},y}(A,c,\phi)\right| \geq \frac{\varepsilon''}{2}\right] < 2 \cdot \exp\left(-2\left(\frac{\varepsilon''}{2}\right)^2 t\right) = \frac{\delta''\xi}{N^{\mathrm{dc}} \cdot \left|\mathrm{TypeNet}_{s,\varepsilon''}\right|}$$

Union-bounding over all density values in the tuple's type, we conclude that the probability that any estimate is $\frac{1}{2}\varepsilon''$-far from the real value, i.e. the probability of failure to output a compatible cluster, is less than $\delta''\xi/\left|\mathrm{TypeNet}_{s,\varepsilon''}\right|$.

We wish to ensure a high enough probability of outputting compatible clusters for most vertices in each cluster induced by $\mathrm{TypeNet}_{s,\varepsilon'}$. Consider some such cluster. The expected fraction of Y tuples from this cluster, for which the estimator outputs an incompatible cluster, is less than $\delta''\xi/\left|\mathrm{TypeNet}_{s,\varepsilon''}\right|$. Applying Markov's inequality to the tuples in this cluster, we conclude that with probability greater than $1-\delta''/\left|\mathrm{TypeNet}_{s,\varepsilon''}\right|$, the estimator outputs a compatible cluster for all but a ξ-fraction of them. We now union-bound again, this time over all clusters in $\mathrm{TypeNet}_{s,\varepsilon''}$, to conclude that with probability greater than $1 - \delta''$, the clustering is indeed correct for all but a ξ-fraction of the tuples in each cluster.

Finally, we must contend with the fact that the estimator does not actually know $\tilde{\mathcal{P}}$. Instead of using the (single) estimator's knowledge of $\tilde{\mathcal{P}}$ to decide which tuples within every sample originate in which partition cell in $\tilde{\mathcal{P}}$, we will have multiple estimators: There will be one estimator for every possible assignment of each subtuple of each $(k-s)$-tuple used in the type estimation sample — each of these estimators assumes knowledge of a different $\tilde{\mathcal{P}}$. Now, the number of possible choices of partition cells for subtuples of a single $(k-s)$-tuple is at most $m^{2^{(k-s)}}$; and over all t tuples, the number $p_{6.3.4}(\varepsilon'', \delta'', \xi)$ of such choices is less than $m^{2^{(k-s)} \cdot t} = m^{O(\ln(1/\delta''\xi) \cdot \mathrm{poly}(1/\varepsilon''))} = \exp(\mathrm{poly}(1/\varepsilon'') \cdot \ln(1/\delta''\xi))$, thus the total number of type estimators is as claimed.

The estimators all share the same sequence U of sampled vertices as the single estimator assuming knowledge of the partition: $t \cdot (k - s) = \mathrm{O}\big(\ln(1/\delta''\xi)\big) \cdot \mathrm{poly}(1/\varepsilon'')$ vertices are sampled, as claimed. □

6.3.3 Distinguishing good and bad partition oracles

We have established that an unknown partition satisfying a certain density characteristic can be replaced with a (large) set of partition oracles of our construction, one of whose induced partitions satisfies the density characteristic approximately. If no approximately-satisfying partition exists, our construction will still yield a set of oracles, but they will be useless — none of them will satisfy the density characteristic even approximately; we need to be able to tell these two cases apart.

Lemma 6.3.5. *Suppose one is given a set S of (q, m, k) partition oracles for a hypergraph $\mathcal{H}$, with shared query complexity q. There exists a probabilistic algorithm making at most $q_{6.3.5}(\varepsilon', \delta, q) = O\Big(\varepsilon'^{-2}\log(1/\delta) \cdot \log(|S|) \cdot q\Big)$ queries to $\mathcal{H}$ for which the following holds:*

- *If one of the oracles' induced partitions $\frac{1}{2}\varepsilon'$-approximately satisfies Ψ, then the algorithm outputs* accept *with probability at least $1 - \delta$.*
- *If none of the oracles' induced partitions ε'-approximately satisfy Ψ, then the algorithm outputs* reject *with probability at least $1 - \delta$.*

Proof. Essentially, we can obtain good estimates of the density characteristic of each oracle, and decide accordingly.

Consider a single oracle $\pi \in S$, inducing partitions $\mathcal{P}^\pi$; our estimate of its density characteristic shall be denoted ψ^U. Set, with foresight,

$$t = 8 \cdot \log(2) \cdot \frac{1}{\varepsilon'^2} \cdot \left(\log\left(\frac{1}{\delta}\right) + \log\left(N^{\mathrm{dc}}\right) + \log(|S|)\right)$$

(recalling that N^{dc} is the number of density values in a density characteristic; see Observation 6.2.12). We sample t sequences of k vertices each: $((x_{h,1}, \ldots, x_{h,k}))_{h=1}^{t}$; let x_h denote the h^{th} k-tuple.

Now, for the partition set vertex density estimates, and for $s \leq k$, we use the first s elements of each sampled tuple to estimate the densities for that arity — we set (abusing notation somewhat)

$$\rho^U(s, j) = \frac{1}{t}|\{h \in [t] \mid \pi(x_h) = j\}|$$

As for the edge density estimates, for every color $c \in \mathcal{C}(H)$ and $\phi \in \Phi^{r(c)}$ we let

$$\begin{aligned}\mu^U(c, \phi) &= \frac{1}{t}|\{h \in [t] \mid x_h \in H(c) \text{ and observes } \phi\}| \\ &= \frac{1}{t}\left|\left\{h \in [t] \;\middle|\; \begin{array}{l} x_h \in H(c) \text{ and} \\ \text{for every } A = (j_1, \ldots, j_s) \in \mathrm{Dom}(\phi),\ \pi(x_h(A)) = \phi(A) \end{array}\right\}\right|\end{aligned}$$

that is, $\mu^U(c, \phi)$ is the fraction of the t samples whose first $r(c)$ elements support an $H(c)$ hyperedge and have sub-tuples which the oracle places in the partition cells indicated by ϕ.

To bound the probability of the estimates being overly far from the actual density values, note that, for every sample-set index j, we have

$$\begin{aligned}\mathbf{Pr}[\pi((x_{h,1},\ldots,x_{h,s}))=j] &= \rho^{\mathcal{P}^\pi}(s,j)\\ \mathbf{Pr}[(x_{h,1},\ldots,x_{h,s})\in H(c) \text{ and observes } \phi] &= \mu^{\mathcal{P}^\pi}(c,\phi)\end{aligned}$$

since the tuple vertices, and hence also the tuples, are sampled uniformly and independently. The estimates $\rho^U(s,j)$ and $\mu^U(c,\phi)$ each admit, therefore, a large deviation bound:

$$\mathbf{Pr}\left[\left|\rho^U(s,j)-\rho^{\mathcal{P}^\pi}(s,j)\right|>\frac{\varepsilon'}{4}\right]<2\cdot\exp\left(-2\left(\frac{\varepsilon'}{4}\right)^2 t\right)=\frac{\delta}{N^{\mathrm{dc}}\cdot|S|}$$

and the bound for $\mu^U(c,\phi)$ is the same. Union-bounding over all N^{dc} density values in the characteristic, we find that with probability greater than $1-\delta/|S|$, our estimates will indeed all be within less than $\varepsilon'/4$ of the correct values, Union-bounding again over all oracles in S, we find that, with probability greater than $1-\delta$, all oracle density characteristic estimates are correct to within less than $\varepsilon'/4$ — independently of which density characteristic these are.

Conditioning on this event, if any of the oracles' partition $\varepsilon'/2$-approximately satisfies Ψ, this oracle's estimate will be at distance under $3\varepsilon'/4$ from Ψ; while if no oracle's partition even ε'-approximately satisfies Ψ, all estimates' distances from Ψ will be higher than $3\varepsilon'/4$. In the former case, we accept, while in the latter, we reject. This completes a valid algorithm meeting the requirement of the claim, with probability of success greater than $1-\delta$.

Finally, the number of (single) oracle invocations in making the estimate is t times the number of subsequences of elements of the k-tuples, which is less than $k!\cdot 2^k$. An additional $t\cdot|\mathcal{C}(H)|$ queries to the hypergraph are made. As the oracles have shared query complexity q, the total number of queries made for estimating all of their density characteristics is: $t\cdot\left(k!\cdot 2^k\cdot q+|\mathcal{C}(\mathcal{H})|\right)=O\left(\varepsilon'^{-2}\log(1/\delta)\cdot q\right)$, as claimed. □

With the ability to generate an appropriate set of oracles, and to distinguish whether any of them induces an acceptable partition, we can now proceed to prove the upper bound:

Proof of Theorem 6.1. Set $\delta=1/6$ and $\varepsilon'=\varepsilon$. Our algorithm acts as follows: The test generates a set S of oracles as described in Lemma 6.3.2, applies the distinguishing algorithm of Lemma 6.3.5 to these oracles, and accepts if and only if the algorithm accepts.

If the hypergraph satisfies Ψ, then by Lemma 6.3.2, with probability at least $5/6$, one of the oracles induces a partition which $\frac{1}{2}\varepsilon$-approximately satisfies Ψ; such an oracle will be accepted by the algorithm of Lemma 6.3.5 with probability at least $5/6$, so with probability at least $2/3$ overall, the test accepts.

If the hypergraph does not ε-approximately satisfies Ψ, then no oracle is such that its induced partition ε-approximately satisfies Ψ, so all the oracles will be rejected with probability at least $5/6$.

The oracles' shared query complexity is $q = \text{poly}(1/\varepsilon) \cdot O(\ln(1/\delta)) = \text{poly}(1/\varepsilon)$, and the number of oracles is $|S| = \exp(\text{poly}(1/\varepsilon)) \cdot \delta^{-O(1/\varepsilon)}$, so the total number of queries made by the distinguishing algorithm of Lemma 6.3.5 is $\text{O}\big(\varepsilon^{-2}\log(|S|) \cdot q\big) = \text{poly}(1/\varepsilon)$.□

6.4 A lower bound on testing partition properties

In this section we show that Theorem 6.1 of the previous section cannot be strengthened from polynomial pseudo-testing to polynomial testing, by the following:

Theorem 6.2. *There exists a density characteristic Ψ for hypergraphs of maximum arity 3, such that testing Π_Ψ requires as many queries as testing a digraph for being triangle-free, up to a constant factor (specifically, $\Omega\big((c'/\varepsilon)^{c'\cdot\ln(c'/\varepsilon)}\big)$ queries are required for some global constant c').*

The combination of the upper bound Theorem 6.1 and this lower bound implies immediately that pseudo-testing is significantly weaker than actual testing:

Corollary 6.3. *The testing query complexity of some partition properties is not a polynomial function of their pseudo-testing query complexity.*

The lower bound Theorem 6.2 will be proven via a reduction (in the sense of Definition 2.4.1) from testing triangle-freeness to testing a partition property which we shall construct. Our construction will use the density characteristic to 'align' a partition of the vertex pairs with a partition of the 2-tuples into edges and non-edges; having done so, we will constrain every 3-tuple to contain at least one pair of vertices which is a non-edge, that is, a pair that resides in the 2-tuple partition cells containing only non-edges. This will make for a straightforward reduction from triangle-freeness testing to testing the satisfaction of the set of density characteristics corresponding to the above constraints.

6.4.1 Expressing basic constraints with density characteristics

To express the constraints necessary for the reduction from triangle-freeness, we shall explore the expressive power of partition properties, gradually establishing its expansion.

The first obvious constraints that we can express using a density characteristic set are the equality of a density value, for single (vertex or edge) density values, e.g. $\mu(c, \phi) = \alpha$, where $\alpha = 0$ means "there are no edges respecting a certain NTD" and $\alpha = 1$ means 'all tuples are edges respecting this NTD". One can also constrain the *sum* of several density values. An important example of this would be $\sum_{\phi\in\Phi^s} \mu(c, \phi) = \alpha$, constraining the total density of the edge relation of color c to be α.

We may also constrain relations of μ values or ρ values to each other, thus expressing the constraint of partition sets having certain sets of equal size, or sizes which are functions of each other.

We would like to make finer and more elaborate constraints regarding the hypergraph edge relations. Efforts in this direction may bear some fruit, e.g.:

Observation 6.4.1. If two (sets of) constraints on hypergraphs (without loss of generality, having the same set of colors) are partition-expressible, then so is their disjunction — using the union of the density characteristic sets expressing each of them (and perhaps promoting first the density characteristics for one of the constraints to a higher value of m, by constraining the gratuitous sets to be empty).

But it may not be possible to achieve much more than the basics described above. However, this section focuses on a lower bound rather than expressivity in general, and for this purpose we may avail ourselves of 'easy' auxiliary relations, added to our hypergraphs, to increase the expressive power using combinations of density constraints. It will later become clear how such relations are useful for our lower bound construction; for now let us describe the mechanism for their use:

A partition cell $X_j^{\mathcal{P},r(c)}$ with respect to $(m,k,\mathcal{C})$-partition functions $\mathcal{P}$ is said to *capture* the color $c \in \mathcal{C}$ if $X_j^{\mathcal{P},s} = H(c)$, i.e. the partition cell contains exactly those $r(c)$-tuples which are edges of color c. A set of partition cells is said to capture c if their union contains exactly those tuples being edges of color c.

Lemma 6.4.2. *Assume $m > 1$. Fix a color c and let $S \subseteq [m]$. There exists a density characteristic set Ψ_1 (respectively, Ψ_2) expressing the constraint of $\left\{X_j^{\mathcal{P},r(c)} \;\middle|\; j \in S\right\}$ capturing $H(c)$ (respectively, capturing $H(c)^{\mathsf{c}} = \prod_{i=1}^{r_c} X \setminus H(c)$).*

Proof. For any $j \in S$, let ϕ_j be the NTD mapping $[r(c)] \overset{\phi}{\mapsto} j$, with ϕ not being defined for any other subsequence of $[k]$. We make the constraints $\mu(c,\phi_j) = \rho(r(c),j)$ for every $j \in S$, and $\mu(c,\phi_j) = 0$ for all $j \in [m] \setminus S$. This ensures that all tuples in each $X_j^{\mathcal{P},r(c)}$ are in $H(c)$, and that prevents any tuples in $H(c)$ from originating in other cells of arity $r(c)$, thus achieving the desired overall constraint.

For capturing $H(c)^{\mathsf{c}}$, we constrain $\mu(c,\phi_j) = 0$ for all $j \in S$, and use the sum constraint $\sum_{j \notin S} \mu(c,\phi_j) = 1 - \sum_{j \in S} \rho(r(c),j)$. □

In essence, the above describes a 'sacrifice' of an edge relation, as it will not hold any 'information' other than our choice of a partition cell, or union of cells, at the appropriate arity. Having made this sacrifice, however, we have increased our expressive power regarding the captured partition cells:

Observation 6.4.3. One may constrain intersections of NTD-respecting tuple sets not merely with edge relations (i.e. constrain the density of $H_\phi(c)$ sets) but also the

intersections of NTD tuple sets with other partition cells (or unions of partition cells). Thus the set

$$\left\{ y \in \bigcup_{j \in S} X_j^{\mathcal{P},s} \;\middle|\; \forall\, B \in \mathrm{Dom}(\phi)\ \left[y(B) \in \phi(B) \right] \right\}$$

can be constrained by adding a color $c_{\phi,S}$ with $r(c_{\phi,S}) = s$, constraining $\left\{ X_j^{\mathcal{P},s} \;\middle|\; j \in S \right\}$ to capture $c_{\phi,S}$, and then using constraints on $H_\phi(c_{s,j})$ (which can be made 'directly' through the density characteristic set).

6.4.2 FOL constraints and density characteristic composition

Definition 6.4.4. For a hypergraph $\mathcal{H}$ with colors $\mathcal{C}(\mathcal{H})$, the *vocabulary* $\tau_{\mathcal{C}}$ consists of a symbol R_c for every color $c \in \mathcal{C}(\mathcal{H})$, with arity $r(c)$, and no constants or function symbols.

Throughout the rest of the section, we refer to formulae and sentences of First-Order Logic without equality, with some fixed vocabulary $\tau_{\mathcal{C}}$; hypergraphs having color set $\mathcal{C}$ are said to *respect* $\tau_{\mathcal{C}}$.

Definition 6.4.5. Consider some partition functions $\mathcal{P}$ of a hypergraph $\mathcal{H}$, some formula $\varphi(x_1, \ldots, x_s)$ and some $S \subseteq [m]$. The set of partition cells $\left\{ X_j^{\mathcal{P},s} \;\middle|\; j \in S \right\}$ is said to *capture* φ if $\bigcup \left\{ X_j^{\mathcal{P},s} \;\middle|\; j \in S \right\}$ contains exactly those s-tuples which satisfy φ.

Definition 6.4.6. Consider a function f from the labeled hypergraphs of order s with color set $\mathcal{C}$ to $\{0, 1\}$. We denote by $\mathcal{D}_f$ the set of all hypergraphs with color set $\mathcal{C} \cup \{c'\}$, and with $r(c') = s$, such that for every $\mathcal{H} \in \mathcal{D}_f$, $H(c')$ contains exactly those tuples $x = (x_1, \ldots, x_s)$ for which f returns 1 when applied to the labeled hypergraph of $\mathcal{H}$ induced by $\{x_1, \ldots, x_s\}$. For such hypergraphs we call f a *deriving function* for color c'. Similarly, for a color set $\mathcal{C}$ and a set of functions $\mathcal{F} = \{f_{c'} \mid c' \in \mathcal{C}'\}$, $\mathcal{D}_{\mathcal{F}}$ is the set of hypergraphs with color set $\mathcal{C} \cup \mathcal{C}'$ for which each $f_{c'}$ is a deriving function for the hypergraph's c' relation.

Definition 6.4.7. A formula $\varphi(x_1, \ldots, x_s)$ (with respect to vocabulary $\tau_{\mathcal{C}}$) said to be *partition-expressible with auxiliary color set* $\mathcal{C}'$ if $\mathcal{C}'$ contains relations of arity at most s, and if there exists an integer m, a set $S \subseteq [m]$, and a set Ψ of $(m, k, \mathcal{C} \cup \mathcal{C}')$ density characteristics, such that the following holds. First, the hypergraphs satisfying Ψ have uniform deriving functions for the colors in $\mathcal{C}'$ — that is, there exists a set of functions $\{f_{c'} \mid c' \in \mathcal{C}'\}$, such that a hypergraph with color set $\mathcal{C}' \cup \mathcal{C}$ satisfies Ψ if and only if it is in $\mathcal{D}_{\mathcal{F}}$. Second, for a hypergraph $\mathcal{H}$ satisfying Ψ, the partitions with which it satisfies Ψ are those in which S captures φ. A formula is said to be *partition-expressible* if there exists an auxiliary color set $\mathcal{C}'$ with which it is partition expressible.

The first requirement for partition-expressibility is of importance to us, as we are considering hypergraphs in which only the $\mathcal{C}$ relations are known, not any auxiliary

relations. With deriving functions, we are able to complete the missing relations using the existing ones.

Observation 6.4.8. If a formula φ (with at least one free variable) is partition-expressible, then so is its negation, with the same number of partition sets per arity and the same auxiliary color set: If Ψ is a density characteristic set expressing the constraint of S capturing φ, then $S^{\mathsf{c}} = [m] \setminus S$ captures $\neg\varphi$ with respect to Ψ, with the same auxiliary color set and deriving functions.

We would ideally like to establish the partition-expressibility of as large a fragment of FOL as possible; we come up against a problem, however, already for mere atomic formulae, before considering connectives or quantifiers: When we capture a relation with a partition cell index (or a set of indices), we are able to set aside those tuples satisfying, say, $R_c(x_1, x_2, x_3)$ or $R_c(x_1, x_3, x_2)$; but what about $R_c(x_1, x_1, x_3)$? Density constraints do not allow us to distinguish tuples with element repetitions. Bearing in mind that our objective is merely expressing triangle-freeness, we shall choose to circumvent the issue and express formulae which are free of such repetition:

Definition 6.4.9. A *repeat-free* FOL formula is one in which no variable appears twice within the tuple of arguments for a relation symbol.

Lemma 6.4.10. *A repeat-free atomic FOL formula $\varphi(x_1, \ldots, x_s)$ (with respect to $\tau_{\mathcal{C}}$) is partition-expressible by a partition with $m = 2$ with an auxiliary color set $\mathcal{C}' = \{c_\varphi\}$, with $r(c_\varphi) = s$.*

Proof. As $\tau_{\mathcal{C}}$ has no function symbols or constants, the repeat-free atomic formulae are all of the form $R_c\big(x_{j_1}, \ldots, x_{j_{r(c)}}\big)$, for some color c, with the j_i's all distinct. (Note, however, that it may be the case that $r(c) < s$, i.e. some variables may be unused.) Fix some such formula φ. By Lemma 6.4.2, there exists a set of density characteristics Ψ_1 (with our choice of $m = 2$ and vocabulary $\tau_{\mathcal{C}\cup\mathcal{C}'}$) constraining $X_1^{\mathcal{P},r(c)} = H(c)$; there similarly exists Ψ_2 constraining $X_1^{\mathcal{P},s} = H(c_\varphi)$.

Now, consider the set $\Phi_\ell = \big\{\phi \in \Phi^s \;\big|\; \big(j_1, \ldots, j_{r(c)}\big) \overset{\phi}{\mapsto} \ell\big\}$. This is set of NTDs is satisfied by those s-tuples whose subtuples corresponding to φ originate in $X_\ell^{\mathcal{P},r(c)}$. We impose the sum constraints $\sum_{\phi\in\Phi_1} \mu(c_\varphi, \phi) = \rho(s, 1)$ and $\sum_{\phi\in\Phi_2} \mu(c_\varphi, \phi) = 0$ (recall that $m = 2$, so $\Phi^k = \Phi_1 \cup \Phi_2$). The combination of these implies that $H(c_\varphi)$ contains exactly the set of s-tuples respecting some NTD from Φ_1. We now conjunct our constraints with those of Ψ_2 (that is, take the intersection of the density characteristic sets), so that the s-tuples in $X_1^{\mathcal{P},s}$ are exactly those respecting some NTD from Φ_1; finally, we conjunct our constraints with those of Ψ_1, so that respecting an NTD in Φ_1 means having $\big(x_{j_1}, \ldots, x_{j_{r(c)}}\big) \in H(c)$ — and the s-tuples in $X_1^{\mathcal{P},s}$ are exactly those with $\big(x_{j_1}, \ldots, x_{j_{r(c)}}\big) \in H(c)$. Thus $X_1^{\mathcal{P},s}$ captures φ exactly (without having imposed any other constraint on other sets $X_j^{\mathcal{P},s}$ for $j \neq 1$).

Finally, a deriving function for c_ϕ would be the function which returns 1 if a hypergraph $\mathcal{H}'$ with vertex set $x_1, \ldots, x_s$ satisfies $\left(x_{j_1}, \ldots, x_{j_{r(c)}}\right) \in H'(c)$, and 0 otherwise.□

Note. The 'formal' number of variables of φ is significant: It is a different thing to express, say, $E(x_1, x_2)$ as a formula of 2 or of 3 variables. In the former case, in fact, one does not even need an auxiliary relation, as the same set of 2-tuples constrained to capture E also captures the atomic formula $E(x_1, x_2)$.

Lemma 6.4.11. *If formulae $\varphi_1(x_1, \ldots, x_{s_1})$ and $\varphi_2(x_1, \ldots, x_{s_2})$ are both partition-expressible with partitions of size m_1 and m_2, respectively, and (disjoint) auxiliary color sets $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, then the formulae $(\varphi_1 \vee \varphi_2)$, $(\varphi_1 \wedge \varphi_2)$ are also partition-expressible, with $m = m_1 \cdot m_2$ and auxiliary color set $\mathcal{C}' = \mathcal{C}_1 \cup \mathcal{C}_2$.*

To prove this, we will require the ability to refine the constraints inducing any set of density characteristics with the constraints inducing any other set:

Definition 6.4.12. Let ψ_1 and ψ_2 be $(m_1, k, \mathcal{C} \cup \mathcal{C}_1)$ and $(m_2, k, \mathcal{C} \cup \mathcal{C}_2)$ density characteristics ($\mathcal{C}_1$ and $\mathcal{C}_2$ are disjoint). The *composition* of the two density characteristics, denoted $\Psi_{\psi_1 \otimes \psi_2}$, is an $(m, k) = (m_1 \cdot m_2, k)$ density characteristic set with respect to the color set $\mathcal{C} \cup \mathcal{C}_1 \cup \mathcal{C}_2$. Abusing our earlier definition somewhat, denote $\mathcal{P}(s) = \mathcal{P}_1 \times \mathcal{P}_2$ and think of the partition functions for an (m, k)-partition as though the $m_1 \cdot m_2$ cells have pairs of indices rather than a single index: $(\mathcal{P}_1 \times \mathcal{P}_2)(s) : \prod_{i=1}^{s} X \to [m_1] \times [m_2]$. Now let $\mathcal{P}_1(s) : \prod_{i=1}^{s} X \to [m_1]$ and $\mathcal{P}_2(s) : \prod_{i=1}^{s} X \to [m_2]$ be the projections of $\mathcal{P}(s)$ onto the first and second coordinates, respectively, i.e. $x \xmapsto{\mathcal{P}(s)} ((\mathcal{P}_1(s))(x), (\mathcal{P}_2(s))(x))$. Now, partition functions $\mathcal{P}$ satisfy $\Psi_{\psi_1 \otimes \psi_2}$ if the projected $\mathcal{P}_1$ and $\mathcal{P}_2$ partition functions satisfy ψ_1 and ψ_2 respectively. In other words, $\Psi_{\psi_1 \otimes \psi_2}$ contains all density characteristics ψ meeting sum constraints on ρ and μ 'gathering' the refined partition cells in an entire cell of $\mathcal{P}_1$ or of $\mathcal{P}_2$. For ρ, these constraints are:

$$\sum_{j_1 \in [m_1]} \rho^{\psi}\left(k', (j_1, j_2)\right) = \rho^{\psi_2}\left(k', j_2\right) \qquad \sum_{j_2 \in [m_2]} \rho^{\psi}\left(k', (j_1, j_2)\right) = \rho^{\psi_1}\left(k', j_1\right)$$

for every $j_1 \in [m_1]$ and $j_2 \in [m_2]$ respectively. For μ values, We need a bit more machinery. Every NTD ϕ' in $\Phi^{k'}$ with respect to $m_1 \cdot m_2$ corresponds to two NTDs $\phi_{\phi',1}, \phi_{\phi',2}$ with respect to m_1 and m_2 respectively, with the same domain as ϕ', such that $\phi'(A) = \left(\phi_{\phi',1}(A), \phi_{\phi',2}(A)\right)$ — the projections of ϕ' onto the first and second coordinates. Now, for some ϕ_1, let Φ'_1 be the set of all NTDs ϕ' in $\Phi^{k'}$ with respect to $m_1 \cdot m_2$ for which $\phi_{\phi',1} = \phi_1$, and let Φ'_2 be defined similarly for any ϕ_2. The sum constraints on $\Psi_{\psi_1 \otimes \psi_2}$ for μ values are:

$$\sum_{\phi' \in \Phi_1} \mu^{\psi}\left(c, \phi'\right) = \mu^{\psi_1}(c, \phi_1)$$

for every $c \in \mathcal{C} \uplus \mathcal{C}_1$ and ϕ_1 in $\Phi^{k'}$ with respect to m_1, and

$$\sum_{\phi' \in \Phi_2} \mu^{\psi}(c, \phi') = \mu^{\psi_2}(c, \phi_2)$$

for every $c \in \mathcal{C} \uplus \mathcal{C}_2$ and ϕ_2 in $\Phi^{k'}$ with respect to m_2.

Definition 6.4.13. Let Ψ_1 and Ψ_2 be $(m_1, k, \mathcal{C} \uplus \mathcal{C}_1)$ and $(m_2, k, \mathcal{C} \uplus \mathcal{C}_2)$ density characteristic sets (with $\mathcal{C}_1$ and $\mathcal{C}_2$ disjoint). The *composition* of the two density characteristics, denoted $\Psi_1 \otimes \Psi_2$, is the set of all compositions of pairs of characteristics from Ψ_1 and Ψ_2, i.e. $\Psi_1 \otimes \Psi_2 = \bigcup_{\psi_1 \in \Psi_1} \bigcup_{\psi_2 \in \Psi_2} \Psi_{\psi_1 \otimes \psi_2}$.

Proof of Lemma 6.4.11. Let Ψ_1, Ψ_2 be the density characteristic sets expressing the two formulae (and their negations), respectively, with capturing cell index sets $S_1 \subseteq [m_1]$ and $S_2 \subseteq [m_2]$ respectively. Consider the composition $\Psi_1 \otimes \Psi_2$ and some partition functions $\mathcal{P}$ with respect to this composition: $x \in \prod_{i=1}^{s} X$ satisfies φ_1 if and only if $x \in X^{\mathcal{P}_1, s}_{(j_1, j_2')}$ for some $j_1 \in S_1$ and some $j_2' \in [m_2]$; $x \in \prod_{i=1}^{s} X$ satisfies φ_2 if and only if $x \in X^{\mathcal{P}_1, s}_{(j_1', j_2)}$ for some $j_2 \in S_2$ and some $j_1' \in [m_1]$. Thus, the composed partition cells with index set $S_1 \times S_2$ capture $(\varphi_1 \wedge \varphi_2)$; and by De-Morgan's law, the cells with index set $(S_1{}^{\mathsf{c}} \times (S_2)^{\mathsf{c}})^{\mathsf{c}}$ capture $(\varphi_1 \vee \varphi_2)$. The expressibility is maintained, as the auxiliary relations with colors in $\mathcal{C}_1$ and $\mathcal{C}_2$ are unaffected by the composition (we simply keep the deriving functions for the relations in both auxiliary relation sets). □

Lemma 6.4.14. *If a formula $\varphi(x_1, \ldots, x_d)$ is partition-expressible with auxiliary color set $\mathcal{C}'$, and with deriving functions $\mathcal{F}$, then there exists density characteristic sets $\Psi_{\varphi,\exists}$ and $\Psi_{\varphi,\forall}$, which are only satisfied by hypergraphs in $\mathcal{D}_{\mathcal{F}}$, and their satisfying graphs are those whose sub-hypergraph obtained by considering the $\mathcal{C}$ relations only, satisfies*

$$\forall\, x_1 \ldots \forall\, x_d \left[\varphi(x_1, \ldots, x_d)\right] \qquad\qquad \exists\, x_1 \ldots \exists\, x_d \left[\varphi(x_1, \ldots, x_d)\right]$$

respectively. In other words, at least one of the sub-hypergraph's s-tuples satisfies φ if the graph satisfies $\Psi_{\varphi,\exists}$, and all of the sub-hypergraph's s-tuples satisfy φ if the graph satisfies $\Psi_{\varphi,\forall}$.

Proof. Constrain a set S of partition cells to capture $\varphi_k(x_1, \ldots, x_d)$; now constrain the set $\bigcup \left\{ X_j^{\mathcal{P},d} \;\middle|\; j \in S \right\}$ to be non-empty (for an $\exists$ constraint) or full (for a $\forall$ constraint), i.e. constrain either $\sum_{j \in S} \rho(d, j) > 0$ or $\sum_{j \in S} \rho(d, j) = 1$. The density characteristic set Ψ for these constraints is indeed a set satisfied by exactly those pairs of a hypergraph $\mathcal{H}$ with auxiliary relations as per the deriving function, with partition functions with which S captures φ, and with a tuple of $\mathcal{H}$ satisfying φ (or with all tuples of $\mathcal{H}$ satisfying φ in the case of a $\forall$ constraint). Thus the two sentences are partition-expressible □

6.4.3 The reduction from testing triangles

Let $\mathcal{C} = \{c\}$ with $r(c) = 2$. The property of the binary relation $H(c)$ being triangle-free (in other words, 3-cycle free) is the property of all hypergraphs which satisfy the following FOL sentence:

$$\varphi_{\text{triangle-free}} = \forall\, x_1, x_2, x_3 \left[\neg(R_{c,2}(x_1, x_2) \wedge R_{c,2}(x_2, x_3) \wedge R_{c,2}(x_3, x_1))\right]$$

with this fact at hand, we can proceed to prove our lower bound.

Note. The formula above forbids degenerate triangles as well, i.e. ones in which two or more of the vertices are the same. Regarding these we can either use the fact that the known lower bound of [AS04a] uses a tri-partite graph with no degenerate triangles, or better still, note that a degenerate triangle must contain a self-loop, while non-degenerate triangles do not contain them; thus if a graph is free from having non-degenerate triangles, then it is $1/n$-close to being altogether triangle-free, and a graph is at least as far from being triangle-free as it is from being non-degenerate-triangle free. Consequently, a test for degenerate-triangle-freeness in digraphs making q queries is a valid test for triangle-freeness for $n = \Omega(1/\varepsilon)$. We may therefore disregard the issue of degenerate triangles.

Proof of Theorem 6.2. By Lemma 6.4.14, $\varphi_{\text{triangle-free}}$ is a partition-expressible constraint, if we add three auxiliary relations of arity 3 (one for each of the relation symbol appearing in the sentence), each of which with a deriving function. Let Ψ denote the density characteristic set guaranteed by the lemma (expressing this constraint using the above-mentioned auxiliary relations) and consider some hypergraph test for Π_Ψ making $q(\varepsilon)$ queries.

Given oracle access to a digraph input with edge set E, we simulate an oracle to a hypergraph with the color set of Ψ, as follows: Queries to $H(c)$ are answered as queries to the digraph; when a query to an auxiliary relation is made about a certain tuple, the oracle queries the subgraph induced by the tuple vertices, and reports whether a hyperedge of the auxiliary relation exists by applying the appropriate deriving function to the (labeled) queried subgraph. If the input digraph is triangle-free, then the simulated hypergraph satisfies Π_Ψ; if the input digraph is ε-far from being triangle-free, then the simulated hypergraph is at least ε-far from Π_Ψ, as, in particular, one must alter at least an ε-fraction of E in order to satisfy $\varphi_{\text{triangle-free}}$.

This oracle meets the requirements of Definition 2.4.1, with $f(\varepsilon) = \varepsilon$, $h(n) = n$ and $g(n) = 9$ (as each query to an auxiliary relation requires at most 3^2 queries to E). The property of testing triangle-freeness is therefore reducible to testing an (arbitrary) hypergraph partition property; we now apply Lemma 2.4.2: Since, by [AS04a], the triangle-freeness of a digraph cannot be tested using less than $(c'/\varepsilon)^{c' \cdot \ln(c'/\varepsilon)}$ for some global constant c', so is the case for hypergraph partition properties (up to a constant factor). □

Chapter 7

Open Questions

Some of the research work resulting in this thesis has fully resolved the questions it had set out to address; other questions were given essential answers with a gap between what has been established and a potential for future improvement or tightening; and others have been given only partial answers indicating a way for future research. All of these, however, bring up additional questions, either regarding their continuation and extension, or on issues only touched upon which may have independent interest. Additionally, some questions reflect objects of the author's research efforts, in the context of the previous chapters, which have not yielded concrete results as of yet. All such question have been concentrated in this chapter, mostly grouped by the chapter which inspired them.

7.1 Natural testing and inflatable properties

Naturalization without canonization. Chapter 3 explores natural tests entirely through the prism of canonical (and more generally, non-adaptive) testing; so much so that it can be seen as a further study of canonical tests rather than of naturalness in testing. What can be said regarding the naturality of non-canonical and possibly adaptive tests? Can such tests be made natural without incurring the double penalty of canonization followed by naturalization of a canonical test?

'Natural' testing with an n-dependent number of queries. What kind of properties have tests whose *number of queries* depends on n, but whose *decision*, in some sense, does not? For example, we might consider a test which accepts if the query results satisfy a sentence in some appropriate logic (e.g. First-Order Logic or Monadic Second-Order Logic, with a vocabulary allowing for unqueried edges).

Note that the above two issues are particularly relevant to the question of natural testing in the sparse graph model, in which non-adaptivity is costly to impose, and where many interesting properties investigated thus far actually have n-dependent query complexity.

The "heredity and inflatability gap" for natural testing. Our test naturalization procedure requires much stronger approximate heredity and approximate inflatability than we can deduce in the reverse direction from the existence of a natural test. Can the requirements be somehow relaxed, or alternatively, can it be shown that naturally-testable properties have stronger approximate inflatability and approximate heredity?

Testing a large graph by testing small subgraphs. Goldreich and Trevisan posed in [GT05] the question of whether any test for a hereditary property can be replaced with merely ensuring that a random *small* induced subgraph (not much larger than the subgraph queried by the original test) has the property — as was originally claimed in [GT03, Proposition D.2]. We've shown that being hereditary and inflatable, or having an original test with one-sided error, is a sufficient condition for this to hold. Are these conditions, or similar ones, also necessary? (Note that this question differs from the previous one, at least in that such a test need not be natural and the tested subgraph size might depend on n.)

The benefit of non-natural testing. Some testable properties have a non-constant-factor gap in query complexity between their adaptive and non-adaptive tests; Such a gap may also exist between natural and n-dependent tests. As with adaptivity, it will be bounded by the penalty of naturalizing the test when at all possible. Can one find specific properties exhibiting such a gap, or 'non-contrived' properties for which there is no gap (similarly to Goldreich and Ron's work in [GR10] regarding adaptivity in tests)?

A more appropriate notion of inflatability. Our choices for the definition of a blowup and of (perfect) inflatability are somewhat arbitrary. For example, the property of being the empty graph is inflatable, but the property of being the complete graph is not — since the clusters in a blowup are empty rather than, say, supporting a clique. Also, the property of being H-free, when H itself is a (generalized) blowup of a smaller graph, is not inflatable. However, these properties are all $(s(\delta), \delta)$-inflatable on the average (even though for the case of subgraph freeness, $s(\delta)$ is exceedingly high). Can one devise a more appropriate, perhaps more relaxed notion of inflatability, which covers such properties as well, while still allowing for naturalization with the same polynomial penalty as in Theorem 3.1? We are uncertain whether one can devise a useful notion of graph blowups under which all such properties would be considered 'perfectly' inflatable. Of course, this is not much of an issue with regard to (s, δ)-inflatability, as at high orders the edges within the clusters have a negligible effect on the distance.

Testability of inflatable graph properties. Alon and Shapira have shown in [AS08a] that any hereditary property is testable (albeit with a prohibitively high query complexity). Is this also the case for properties which are only known to be inflatable?

That is, can one use the closure to blowups, rather than the closure to taking induced subgraphs, to devise a test? Perhaps Goldreich and Avigad's recent work in [AG11] can shed some light on this question.

7.2 Hard properties and complexity hierarchies

Hard functions with a combination of desirable features. The two constructions of hard properties in Chapter 4, namely, in Section 4.2 and Subsection 4.6.1, immediately beg the question of whether one can combine the desirable features of two or all three of the constructed properties. Specifically, are there hard graph properties (requiring $\Omega(n^2)$ queries) which are

- both monotone and decidable in **PTIME**?
- monotone, and with a test whose running time is polynomial in n?

Note that one-sided-error testing is a feature of all hard properties, since reading the entire input constitutes a one-sided test with a minimum number of queries up to a constant. Also, it seems likely that the use of an **NPTIME**-decidable small sample space for constructing a hard-to-test property, as in [GGR98, Proposition 10.2.3.2], can yield a monotone property decidable in **NPTIME** at least.

Complexity hierarchies with a combination of desirable features. Assuming that appropriate hard properties can be constructed, is it also the case that the desirable feature of the three query complexity hierarchy results can be combined? Specifically, for any reasonable $q(n)$, is there a dense graph property requiring $\Theta(q(n))$ queries which

- is both monotone and one-sided-testable with $\Theta(q(n))$ queries?
- is monotone, and has a test making $\Theta(q(n))$ queries with running time polynomial in $q(n)$?
- has a $\Theta(q(n))$-query, poly$(q(n))$-time test which is also one-sided?

Towards this end, it may be useful to consider whether one can use a permutation-invariant LDPC code in the initial construction (see Subsection 4.2.2).

Decoupling the dependence on n and ε. For the case of generic functions, Chapter 4 establishes the existence of properties with query complexity $c \cdot q(n) + f(\varepsilon)$ where c is independent of ε. Can this be established in other models? A discussion of this possibility for the case of bounded-degree graphs (with no answer) can be found in the conclusion of Section 4.4. What about properties of dense graphs? This question can be asked, of course, for any combination of the desirable features in the different hierarchy theorems.

Hard properties that are 'self-similar' at different values of n. The constructions in Chapter 4 make no guarantees regarding the relation between Π_{n_1} and Π_{n_2} for

$n_1 \leq n_2$ — not even if $n_1 = n_2 \pm 1$. Can hard properties be constructed, and hierarchies be shown to exist, for properties in which the property is 'similar' for different values of n? e.g. where adding or removing a vertex from a satisfying graph puts it at a relatively small distance from the property? This question can be asked also for any combination of the desirable features in the different hierarchy theorems.

Tighter bounds on the effect of graph blowups on distances. As mentioned in Chapter 3 and Chapter 4, the distance between graphs does not change overmuch when applying an exactly-balanced blowup: It does not increase (an easy observation) and does not drop by a factor higher than 3 (the result of [Pik10, Lemma 14]). An example by Arie Matsliah shows that the distance can drop to as low as 10/11 of the original distance. It would be interesting to tighten both the upper and lower bound on the potential drop in distance, and to gain a better understanding of this drop in distance.

The effect of hypergraph blowups on distances. Does Pikhurko's result regarding the preservation of distance under blowup carry to hypergraphs? Also, what about an extension to hypergraphs of the similar lemma for the case of dispersed graphs and imperfectly-balanced blowups (Lemma 4.5.13)? One tends to believe that both of these should hold. Establishing the latter should also allow proving hierarchy theorems for hypergraphs, or any dense structure.

7.3 Partite and multi-colored dense structures

Subgraph-freeness testing in partite vs general graphs. The state-of-the-art lower bounds on induced subgraph freeness testing (specifically, triangle testing) are based on using the arithmetic-progression-free set constructions in a partite graph (tri-partite for the case of triangles); the fixed number of parts is what allows us to apply this lower bound to the case of colored bipartite graphs or matrices, which can simulate a higher number of parts — but not a general graph. Do better lower bounds hold for testing induced subgraph freeness in general graphs, rather than for testing in partite ones? Can constructions rely specifically on the "non-partiteness" of a graph? It seems that this can be answered negatively, so that lower bounds in the general settings are translatable to partite graph and colored bipartite graph lower bounds (by methods similar to those used in Chapter 5), and partite graph tests can translate to tests of general graphs (through the partitioning of general graphs and the testing of partite subgraphs).

Expressive power of subgraph-freeness with multiple colors. The results presented in this thesis mean that three colors are more expressive than two in bipartite graphs, in that properties which are harder to test can be expressed as freeness of

certain induced subgraphs. What about three-vs-two colors in three-dimensional tensors (i.e. 3-partite 3-uniform hypergraphs)? Perhaps it can it be shown that tensors can be carved up into test-identifiable regions, so as to simulate additional colors (in which case the query complexity of testing freeness of an arbitrary family of subtensors will not be higher when allowing multiple cell colors). Also, is there additional such expressive power in allowing more than three different colors in bipartite graphs?

7.4 Hypergraph partition properties

In studying tuple partition properties, the initial hope was to obtain a proof that a wider class of tuple partition properties is pseudo-testable — rich enough to essentially capture the property of a hypergraph having a certain *regular partition*. To express the constraints necessary for representing such a regular partition, it is necessary to cross-constrain elements and subtuples of a single tuple (see discussion in Section 6.1). Attempts to establish the pseudo-testability of such properties have not met with success thus far; had they succeeded, a test for a regular partition would be at hand, due to the following lemma, which we present informally and without proof here:

Lemma 7.4.1. *Consider the property of a (uniform) hypergraph having an ε-regular partition with a fixed maximum number of partition sets m'. If a hypergraph has a partition with m' sets, whose densities (with respect to the appropriately expressive definitions of partition densities) are close to those densities corresponding to a regular partition, then the graph is $f(\varepsilon)$-close to having a $g(\varepsilon)$-regular partition with m' sets.*

Efficiently testing for regular partitions in hypergraphs. Is the specific property of a hypergraph having a regular partition, with a fixed maximum number of partition sets, testable with $\text{poly}(1/\varepsilon)$ queries? If not, what lower and upper bounds can one establish for the query complexity of this property? We note that the super-polynomial lower bound, established for testing a partition property even with limited expressibility, does not necessarily apply to this particular property.

Pseudo-testing vs. actual testing of rich-constraint partition properties. With the limited expressibility imposed in this work, we've shown that testing a hypergraph partition property is harder than pseudo-testing it. Does this hold for rich-constraint partition properties? That is, can one show that pseudo-testing is, say, polynomial in ε? Or more generally, establish that the query complexity of pseudo-testing is $q_1(\varepsilon, n)$ and find a rich-constraint partition property requiring $q_2(\varepsilon, n)$ queries with $q_1 = o(q_2)$?

Possible hardness of non-rich partition properties. We've established that hypergraph partition properties, even without 'rich' constraints, can capture a property with query complexity super-polynomial in ε. But this construction was not overly

complex, and only utilized a maximum arity of 3. Can a more involved construction be shown to require a super-polynomial number of queries in $1/\varepsilon$, significantly higher than the bound due to the reduction from triangle-freeness testing? We have not even ruled out the possibility that there exist partition properties of this kind whose query complexity must depend on n: This would seem unlikely, as a small random subgraph should exhibit about the same partition as the large graph, and such properties are clearly inflatable; but this is not much more than intuition.

7.5 Expanding the testing model via 'plugging'

Expanding the testing model via 'plugging' testable relations and functions Consider questions of the following type: "Let E' be all vertex tuples of a certain arity, which satisfy a certain condition. Now, given a set of tuples, what fraction of it intersects E'?" or "what is its distance from E'?". One can think of this as a "formula-type property" rather than a "sentence-type property" as in formal logic. Now, suppose one has an oracle which answers questions of this type with certain query complexity to the input structure. It would be interesting to consider property tests which use such oracles as subroutines; in the case of such a subroutine giving an "is in E' / is not in E'" answer, one could think of the test having temporarily or locally added a new edge relation to the structure (in the same way as when quantifying over a relation variable in formal logic, it is used as relation symbols from the vocabulary would be used).

In fact, this is done implicitly by many tests in the literature and some in this work, e.g. when obtaining an approximate clustering of vertices using a signature (Algorithm 4.3 in Chapter 4). One could think of such a test as constructing or learning a probably-approximately-correct partition function, and then applying another test to a structure which has both an edge relation and a partition function. If the construction is valid and the richer-vocabulary structure test is valid, then so is the test of the original structure. This conceptual approach links different testing models in a more general way than mere reductions (Definition 2.4.1), and its study may yield some "meta-results" regarding testing. Thus when given a property whose query complexity is to be ascertained, one could approach the problem by augmenting the input structure with "testable relations" or "testable functions", and only need to consider the modified problem as though these relations or functions were provided perfectly rather than through a test.

7.6 Ordered structures

The dense structure testing models studied in this work all share the requirement that properties be invariant to permutations of the vertices — that is, that properties not relate to any ordering of the vertices. (An alternative definition of a test is proposed in this work — Definition 2.1.4 — explicitly adopting the implications of this fact.)

Testing models in which vertices *are* ordered, with no permutation possible, have not been the object of much study thus far.

Efficient testing of induced (ordered) submatrices. [AFN07] show that it is easy to test a matrix for being free of a fixed set of small submatrices and their permutations. What about a set of forbidden submatrices not closed to permutations? The answer to this question regards testing matrices without ignoring the coordinate order. As part of the research work leading to this thesis, efforts were made to apply the upper bounds of [AFN07] in this context, using a 'conditional regularity' lemma for forbidden small submatrices (see Section 5.1); unfortunately, these efforts have not met with success. On the other hand, there seems to be indication against the unordered-case result carrying to ordered matrices.

Applicability of unordered results to the ordered settings. Generalizing the previous questions, which results carry over from the unordered to the ordered-vertex setting? Some can be seen to easily carry over, such as lower bounds on testing forbidden subgraph freeness — using closure under permutations and a reduction to the unordered case. What about results such as regularity-based (and other) upper bounds? Canonization, adaptivity gaps, etc.? Also, what kind of upper and lower bounds can one obtain in the ordered setting for specific properties with known n-dependent query complexity in the unordered setting?

Bibliography

[ABI86] Noga Alon, Laszlo Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4):567–583, 1986.

[ADL+94] Noga Alon, Richard A. Duke, Hanno. Lefmann, Vojtech Rödl, and Raphael Yuster. The algorithmic aspects of the regularity lemma. *Journal of Algorithms*, 16:80–109, 1994.

[AFKS00] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20:451–476, 2000.

[AFN07] Noga Alon, Eldar Fischer, and Ilan Newman. Efficient testing of bipartite graphs for forbidden induced subgraphs. *SIAM Journal on Computing*, 37(3):959–976, 2007.

[AFNS09] 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. An earlier version appeared in the proceedings of the 38$^{\text{th}}$ STOC, 2006.

[AG11] Lidor Avigad and Oded Goldreich. Testing graph blow-up. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 156–172. Springer, 2011.

[AK99] Noga Alon and Michael Krivelevich. Testing k-colorability. *SIAM Journal on Discrete Mathematics*, 15:211–227, 1999.

[AKKR08] Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics*, 22:786–819, 2008.

[Alo99] Noga Alon. Private communication, 1999.

[Alo02] Noga Alon. Testing subgraphs in large graphs. *Random Structures and Algorithms*, 21(3-4):359–370, 2002.

[AS04a] Noga Alon and Asaf Shapira. A characterization of easily testable induced subgraphs. In *Proceedings of the 15th SODA*, pages 942–951, 2004.

[AS04b] Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer Systems Science*, 69(3):354–382, 2004.

[AS05] Noga Alon and Asaf Shapira. Every monotone graph property is testable. In *Proceedings of the 37th STOC*, pages 128–137, New York, NY, USA, 2005. ACM Press.

[AS06] Noga Alon and Asaf Shapira. A characterization of easily testable induced subgraphs. *Combinatorics, Probability and Computing*, 15(6):791–805, 2006. An earlier version appeared in the proceedings of the 15$^{\text{th}}$ SODA, 2004.

[AS08a] 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. An earlier version appeared in the proceedings of the 46$^{\text{th}}$ FOCS, 2005.

[AS08b] Noga Alon and Asaf Shapira. A separation theorem in property testing. *Combinatorica*, pages 261–281, 2008.

[ASE92] Noga Alon, Joel H. Spencer, and Paul Erdős. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., 1992.

[BCL$^+$06] Christian Borgs, Jennifer Chayes, László Lovász, Vera T. Sós, Balázs Szegedy, and Katalin Vesztergombi. Graph limits and parameter testing. In *Proceedings of the 38th STOC*, pages 261–270, New York, NY, USA, 2006. ACM Press.

[Beh46] Felix A. Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences of the USA*, 32:331–332, 1946.

[BEKKR10] Ido Ben-Eliezer, Tali Kaufman, Michael Krivelevich, and Dana Ron. Comparing the strength of query types in property testing: The case of testing k-colorability. In Goldreich [Gol10], pages 253–259.

[BLR90] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the 22nd STOC*, pages 73–83, New York, NY, USA, 1990. ACM Press.

[BOT02] Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *Proceedings of the 43th FOCS*, 2002.

[BT04] Andrej Bogdanov and Luca Trevisan. Lower bounds for testing bipartiteness in dense graphs. *Proceedings of CCC 2004*, pages 75–81, 2004.

[Elk11] Michael Elkin. An improved construction of progression-free sets. *Israeli Journal of Mathematics*, 184:93–128, 2011.

[Fis04] Eldar Fischer. The art of uninformed decisions: A primer to property testing. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, volume 1, pages 229–264. World Scientific Publishing, 2004.

[FM06] Eldar Fischer and Arie Matsliah. Testing graph isomorphism. In *Proceedings of the 17th SODA*, pages 299–308, New York, NY, USA, 2006. ACM Press.

[FMS07] Eldar Fischer, Arie Matsliah, and Asaf Shapira. Approximate hypergraph partitioning and applications. In *Proceedings of the 48th FOCS*, pages 579–589, 2007.

[FN01] Eldar Fischer and Ilan Newman. Testing of matrix properties. In *Proceedings of the 33rd STOC*, pages 286–295, New York, NY, USA, 2001. ACM Press.

[FN07a] Eldar Fischer and Ilan Newman. Testing of matrix-poset properties. *Combinatorica*, 27(3):293–327, 2007.

[FN07b] Eldar Fischer and Ilan Newman. Testing versus estimation of graph properties. *SIAM Journal on Computing*, 37(2):482–501, 2007.

[Fox11] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174(1):561–579, 2011. available at the following URL: `http://math.mit.edu/~fox/paper-removal.pdf`.

[FR07] Eldar Fischer and Eyal Rozenberg. Lower bounds for testing forbidden induced substructures in bipartite-graph-like combinatorial objects. In *Proceedings of RANDOM 2007*, pages 464–478, Berlin, Heidelberg, 2007. Springer-Verlag.

[FR11] Eldar Fischer and Eyal Rozenberg. Inflatable graph properties and natural property tests. In *Proceedings of RANDOM 2011*, pages 542–554, Berlin, Heidelberg, 2011. Springer-Verlag. Full

version available at http://www.cs.technion.ac.il/~eyalroz/publications/FR2011.pdf.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[GKNR10] Oded Goldreich, Michael Krivelevich, Ilan Newman, and Eyal Rozenberg. Hierarchy theorems for property testing. In Goldreich [Gol10], pages 289–294.

[Gol10] Oded Goldreich, editor. *Property Testing - Current Research and Surveys (outgrow of a workshop at the Institute for Computer Science (ITCS) at Tsinghua University, January 2010)*, volume 6390 of *Lecture Notes in Computer Science*. Springer, 2010.

[Gow07] William Timothy Gowers. Hypergraph regularity and the multidimensional szemerédi theorem. *Annals of Mathematics*, 166(3):897–946, 2007.

[GR99] Oded Goldreich and Dana Ron. A sublinear bipartitness tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999.

[GR02] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. An earlier version appeared in the proceedings of the 29$^{\text{th}}$ STOC, 1999.

[GR07] Mira Gonen and Dana Ron. On the benefits of adaptivity in property testing of dense graphs. In *Proceedings of RANDOM 2007*, pages 525–539, Berlin, Heidelberg, 2007. Springer-Verlag.

[GR09] Oded Goldreich and Dan Ron. On proximity oblivious testing. In *Proceedings of the 41st STOC*, pages 141–150, New York, NY, USA, 2009. ACM.

[GR10] Oded Goldreich and Dana Ron. Algorithmic aspects of property testing in the dense graphs model. In Goldreich [Gol10], pages 295–305.

[GT03] Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, 23(1):23–57, 2003.

[GT05] Oded Goldreich and Luca Trevisan. Errata for [GT03], 2005. available at the following URL:
http://www.wisdom.weizmann.ac.il/~oded/PS/tt-err.ps.

[GW10] Ben Green and Julia Wolf. A note on elkin's improvement of behrend's construction. In David Chudnovsky and Gregory Chudnovsky, editors, *Additive Number Theory*, pages 141–144. Springer, 2010.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[Ish09] Yoshiyasu Ishigami. A simple regularization of hypergraphs, 2009. available from the following URL: `http://arxiv.org/abs/math/0612838`.

[KKR04] Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, June 2004.

[NRS06] Brendan Nagle, Vojtěch Rödl, and Mathias Schacht. The counting lemma for regular k-uniform hypergraphs. *Random Structures & Algorithms*, 28(2):113–179, 2006.

[Obr11] Kevin Obryant. Sets of integers that do not contain long arithmetic progressions. *The Electronic Journal of Combinatorics*, 18:59–73, 2011. available at the following URL: `http://www.emis.ams.org/journals/EJC/Volume_18/PDF/v18i1p59.pdf`.

[Pik10] Oleg Pikhorko. An analytic approach to stability. *Discrete Mathematics*, 310(21):2951 – 2964, 2010.

[PRR03] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Testing membership in parenthesis languages. *Random Structures and Algorithms*, 22(1):98–138, 2003.

[Ron01] Dana Ron. Property testing (a tutorial). In Sanguthevar Rajasekaran, Panos M. Pardalos, John H. Reif, and José D. P. Rolim, editors, *Handbook of Randomized Computing*. Kluwer Press, 2001.

[Ron10] Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2010.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[Sha04] Ronen Shaltiel. Recent developments in explicit constructions of extractors. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, volume 1, pages 229–264. World Scientific Publishing, 2004.

[Sha06] Asaf Shapira. *Graph Property Testing and Related Problems*. PhD thesis, Tel Aviv University, 2006.

[Sze78] Endre Szemerédi. Regular partitions of graphs. In M. Las Vergnas J. C. Bermond, J. C. Fournier and D. Sotteau, editors, *Proc. Colloque Inter. CNRS*, pages 399–401, 1978.

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In 18^{th} *FOCS*, pages 222–227, 1977.

בדיקת־דמה של תכונות חלוקה של רשומות בגרפי־על

שתי תוצאות אחרונות עוסקות בשאלת האפשרות להרחיב את המחלקות הידועות של תכונות גרפי־על, מרובי יחסי קשתות, אשר ניתנות לבדיקה יעילה (במספר שאילתות פולינומי ב־$1/\varepsilon$). בפרט, אנו מעיינים בהכללה של תכונות חלוקה של גרפים: תכונות אשר מסתפקות על־ידי גרף כאשר ישנה חלוקה של צמתיו, אשר מספקת אילוצים הן אודות גדלי הקבוצות הן אודות צפיפות הקשתות בין קבוצות שונות. ידוע כי תכונות במחלקה זו ניתנת לבדיקה יעילה לגבי גרפים; ומחקר קודם הראה כי הכללה בסיסית של תכונות כאלה לגרפי־על ניתן לבדיקה יעילה.

אנו בוחנים הכללה בעלת כושר־ביטוי רב יותר, בה החלוקה אינה רק של הצמתים, אלא גם חלוקה של רשומות הצמתים (זוגות סדורים, שלשות סדורות וכן הלאה עד k־יות סדורות). אנו מראים כי התכונות במחלקה זו – אפילו כאשר מגדירים אותה באופן המגביל את כושר ביטויה (לדוגמה, באופן אשר אינו מאפשר כנראה להביע בתכונה כזו את היותו של גרף־על בעל חלוקה רגולרית, בהתאם ללמת הרגולריות לגרפי־על) – אינן ניתנות ככלל לבדיקה יעילה. מאידך, אנו מראים כי ניתן לבצע לתכונות כאלה 'בדיקת־דמה' יעילה, הווה אומר, ניתן להבחין ביעילות האם קיימות לגרף־העל הנבדק חלוקות המקיימות בקירוב את אילוצי הצפיפות. אלא מאי, 'בדיקת־דמה' זו, המספיקה במקרה של גרפים לעריכת בדיקה של־ממש, אינה מספיקה במקרה הכללי, כפי שמוכיח החסם התחתון על סיבוכיות השאילתות לבדיקה יעילה אותו אנו מציגים.

היראַרכיות סיבוכיות שאילתות עבור גרפים צפופים ומבנים נוספים

מבדיקות טבעיות אנו פונים לחקרן של בדיקות אשר תלויות תלות משמעותית בגודל גרף הקלט שלהן. האם קיימות תכונות כאלה לכל פונקציית תלות שרירותית ב־n, גודל הגרף? אנו עונים לשאלה זו בחיוב, באמצעות הוכחת קיומן של היראַרכיות של מחלקות תכונות לפי סיבוכיות השאילתות שלהן, הן לדגם הצפוף הן לדגם הדליל של בדיקת תכונות גרפים. בקווים כלליים, אנו מוכיחים כי לכל פונקציה סבירה $q(n)$, קיימת תכונה של גרפים אשר לא ניתן לבדוק במספר שאילתות $o(q(n))$ (נמוך משמעותית מ־$q(n)$), אך ניתן לבדוק במספר שאילתות $O(q(n))$.

בדגם הדליל, אנו מבססים משפט היראַרכיה באמצעות תכונה שאינה מלאכותית, קלה לניסוח לכל $q(n)$: תכונת היותו של גרף צביע ב־3 צבעים (הווה אומר, צביע כך שאף קשת אינה מחברת צמתים בצבע שונה), ובעל רכיבים קשירים בגודל $q(n)$ לכל היותר; לתכונה זו ישנה בדיקה ב־$O(q(n))$, שהינה בעלת שגיאה חד־צדדית.

בדגם הצפוף, אנו מוכיחים, למעשה, שלושה משפטי היראַרכיה שונים, אשר כל אחד מהם מוסיף מאפיינים מעניינים של התכונה או של הבדיקה:

- היראַרכיה של מחלקות סיבוכיות שאילתות עבור תכונות אשר ניתנות להכרעה דטרמיניסטית בזמן פולינומי בגודל (כשפות של גרפים), ואשר אלגוריתם הבדיקה שלהן פועל בזמן פולינומי במספר השאילתות.
- היראַרכיה של מחלקות סיבוכיות שאילתות עבור תכונות מונוטוניות (אם כי לאו דווקא כאלה הניתנות להכרעה דטרמיניסטית בזמן פולינומי).
- היראַרכיה של מחלקות סיבוכיות שאילתות עבור תכונות אשר ניתנות להכרעה דטרמיניסטית בזמן פולינומי בגודל (כשפות של גרפים), ואשר הבדיקה שלהן הינה בעלת שגיאה חד־צדדית.

חסמים תחתונים בבדיקה של מבנים צדדיים

מבנה צדדי הוא מבנה אשר צמתיו מתחלקים למספר קבוצות המכונות "צידי" הגרף. כך, לדוגמה, בגרף דו־צדדי ישנן שתי קבוצות צמתים, וכל קשת נמתחת בין צומת מקבוצה אחת לצומת בקבוצה השניה – אין קשתות "בתוך אותו צד".

אנו עוברים מהתמקדות בתכונות של גרפים להתמקדות בתכונות של מבנים צדדיים צפופים. אנו מציגים חסמים תחתונים לבדיקת גרפים דו־צדדיים בעלי קשתות מרובות־צבעים, וכן לגבי גרפי־על k־צדדיים ואחידים־k בקשתותיהם (עליהם ניתן לחשוב כעל מטריצות וטנסורים מעל שדות סופיים קבועים, בהתעלם מסדרם היחסי של שיעורי המטריצה או הטנסור במימדים השונים). תוצאה חיובית קודמת, אשר מתבססת על עקרון של 'רגולריות מותנית' בגרפים דו־צדדיים, ביססה את היכולת לבדוק את היעדרותם של תתי־מבנים קטנים אסורים בגרפים דו־צדדיים, בבדיקה במספר שאילתות שהינו פולינומי בפרמטר $1/\varepsilon$. אנו מוכיחים כי תוצאה זו אינה תקפה כאשר לקשתות ייתכנו מספר צבעים, או כאשר המימד מוגדל במעבר לגרפי־על k־צדדיים ואחידים־k עבור $k \geq 3$; זאת באמצעות חסמים תחתונים (חסמים מלרע) בפונקציה על־פולינומית בפרמטר $1/\varepsilon$. חסמים אלו מהווים פתרון לשאלה פתוחה של אלון, פישר ונוימן.

חישוב זהה עבור גרפים בכל גודל.

כשם שניתן להבחין באיפיונים של בודקים, כך ניתן להגדיר גם אפיונים של תכונות ומחלקות של תכונות. האיפיונים המתבקשים ביותר נוגעים כמובן לסיבוכיות שאילתות: עבור פונקציה $q(n,\varepsilon)$ כלשהיא, נתעניין בתכונות שלהן בדיקה אשר מבצעת $O(q(n,\varepsilon))$ שאילתות (מספר שאילתות שהינו חסום, אסיסמפטוטית, על־ידי קבוע כפול ב־q). דוגמה אחרת למאפיין של תכונות, אשר זכה לעיון מחקרי, הינה הנורשות: תכונה של מבנים צפופים היא נורשת, אם כל תת־מבנה של מבנה בתכונה, המושרה על־ידי תת־קבוצה של צמתי המבנה הגדול יותר, מקיים בעצמו את התכונה.

חיבור זה מכוון להעמיק את הבנתנו את הדגם הצפוף לבדיקת תכונות. זאת, הן לגבי בדיקת תכונותיהם של מבנים מסוגים מסוימים במסגרתו ־ כגון גרפים פשוטים וגרפים דו־צדדיים מרובי־צבעים ־ הן באופן כללי. זאת בין היתר באמצעות פיתוח של סוגי ניפוח של מבנים צפופים; פיתוח איפיונים מועילים חדשים של תכונות בדגם הצפוף; ויישום וקישור של כל אלו למושגים ואיפיונים ידועים בקשר לבדיקת תכונות.

התוצאות שבחיבור, אשר נסקרות להלן, הופיעו ברובן במאמרים, אשר פורסמו בידי המחבר ושותפיו למחקר בכנסים ובכתבי־עת שונים במהלך תקופת מחקר הדוקטורט של המחבר. גרסאותיהם העדכניות ביותר של הפרסומים הללו הינן: [FR07], [GKNR10] ו־[FR11].

תכונות בנות־ניפוח ובדיקות תכונה טבעיות

אנו מתמקדים תחילה בגרפים צפופים, ומעיינים בבדיקות טבעיות שלהם. אנו מציגים מאפיין חדש, היותן של תכונות בנות־ניפוח: תכונה היא בת־ניפוח אם היא סגורה לפעולת ניפוח (מאוזן) של גרפים. אנו מראים כי סיבוכיות השאילתות של בדיקות טבעיות של תכונה קשורה קשר קרוב למידה בה התכונה הינה נורשת־בקירוב ובת־ניפוח־בקירוב. אנו מראים בפרט כי בדיקות לתכונות, שהינן כמעט נורשות וכמעט בנות־ניפוח, ניתן להפוך לבדיקות טבעיות מבלי לשלם מחיר גבוה מאוד במספר השאילתות; זאת על בסיסה של שיטת ה'קנוניזציה' הידועה של בדיקות גרפים צפופים. בכיוון ההפוך, אנו מראים כי תכונות להן בדיקה טבעית הינן נורשות־בקירוב ובנות־ניפוח־בקירוב, במידות התלויות במספר השאילתות של הבדיקה.

באמצעות שיטת ה'אקלום', אותה אנו מפתחים, ההופכת בדיקות לטבעיות, אנו משחזרים חלקית תוקף טענה של גולדרייך וטרביסאן אודות בדיקת תכונות נורשות, ובאמצעותה מכלילים את הקשר בין חסם על בדיקה חד־צדדית ודו־צדדית של היעדר משולשים בגרפים. אנו גם מוכיחים הוכחה מפורשת פשוטה של חיזוק מזערי של החסם התחתון המיטבי הידוע על בדיקה כזו. אנו מעיינים גם ביחסים בין מושג ברוּת־הניפוח למאפיינים אחרים של תכונות ושל בדיקות תכונה בדגם הצפוף, אשר נדונו זה־מכבר במחקר, כגון: שגיאה חד־צדדית, נורשות, ובדיקה בחוסר־מודעות למרחק.

לבסוף, אנו מכלילים את שיטות ה'קנוניזציה' וה'אקלום' למבנים צפופים כלליים שאינם גרפים, ובאמצעותן מבססים את טענותינו לדגם הצפוף בכלל ולא רק לגבי גרפים.

תקציר

עניינו של חיבור זה הינו הבדיקה ההסתברותית של תכונות של מבנים צירופיים (קומבינטוריים). בעיונים בתחום סיבוכיותם של חישובים, חלק ניכר מן המחקר עוסק בשאלות מן הסוג: "כמה משאבים חישוביים מסוג מסויים נדרשים על מנת לפתור בעיה חישובית?" חישובים שהינם בדיקת תכונות מתבססים על שאלת מספר מצומצם (ככל האפשר) של שאילתות לגבי מבנה צירופי כלשהוא, מבלי לקרוא את כולו בגישה ישירה; על אלגוריתם הבדיקה להכריע על־בסיס התשובות לשאילתות האם המבנה מקיים תכונה כלשהיא, או שמא הוא רחוק מלקיים אותה. (אם המבנה אינו רחוק מקיומה של תכונה, מספר קטן של שאילתות עלול לא לגלות את ההבדל בינו לבין מבנה המקיים את התכונה.) על ההכרעה להיות נכונה בהסתברות גבוהה.

המחקר בתחום בדיקת התכונות מתמקד במספר השאילתות הדרוש כמשאב חישובי, ובוחן את סיבוכיות השאילתות של תכונות מסוימות ושל מחלקות של תכונות. זאת, כתלות בגודל המבנים הנבדקים (המסומן ב־n), וכן כתלות בפרמטר המרחק בו מותרת טעות בבדיקה (המסומן ב־ε).

ישנם דגמים (מודלים) שונים של בדיקת תכונות. ההבדלים בין דגמים הינם, בעיקר, בשאלות: אילו מין מבנים עוברים בדיקה; אילו שאילתות ניתן לשאול לגבי המבנה הנבדק; וכן – מה פירושו של מבנה 'רחוק', הווה אומר, מהי פונקציית המרחק (המטריקה) בין מבני קלט אפשריים.

הדגם אשר היווה את מושא המחקר השכיח ביותר בתחום בדיקת התכונות מכונה "הדגם הצפוף", או במקור "הדגם הצפוף לבדיקת גרפים". דגם זה אכן הוגדר לראשונה עבור גרפים, בלתי־מכוונים ופשוטים (חסרי לולאות עצמיות), בהם ייתכנו כל הקשתות האפשריות – מגרפים ריקים ועד מלאים, בני $\binom{n}{2}$ קשתות; כך, שאילתה בדגם זה הינה "האם קיימת קשת בין הצומת ה־i לצומת ה־j בגרף?". המרחק בין שני גרפים הינו מספר הקשתות אשר יש להוסיף או להסיר לאחד מהם כדי לקבל את הגרף השני; בחירה זו של פונקציית מרחק מזכה את הדגם בכינוי "צפוף", שכן גרף בעל מספר תת־ריבועי של קשתות $o(n^2)$ (כגון עץ) הוא קרוב כרצוננו (עבור ערכי n גבוהים מספיק) לגרף ריק. הדגם הצפוף רלבנטי עוד למבנים רבים אחרים: גרפים מכוונים או בעלי קשתות צבועות, גרפי־על (היפרגרפים), מטריצות וטנסורים (או מערכים דו־מימדיים ורב־מימדיים).

גם במסגרת דגם יחיד לבדיקת תכונות, ניתן לדוק ולאפיין אפיונים שונים את אלגוריתמי הבדיקה, להם משמעות לעניין יכולותיהם החישוביות וסיבוכיות השאילתות אשר הם דורשים בהקשרים שונים. לדוגמה, בדיקת תכונה עשויה להיות בעלת שגיאה חז־צדדית: בדיקה כזו לעולם לא תדחה את המבנה הנבדק כאשר המבנה מקיים את התכונה, ותוכל לדחות רק מבנים שאינם מקיימים אותה. תכונה אחרת של בודקים הינה טבעיות: בדיקת תכונה מכונה בדיקה טבעית אם פעולתה הינה בלתי־תלויה לחלוטין בגודלו של הגרף הנבדק, הווה אומר, אלגוריתם הבדיקה שואל מספר שאילתות זהה ומבצע

המחקר בוצע בהנחייתו של פרופסור אלדר פישר, בפקולטה למדעי המחשב.

רוב התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי־עת במהלך תקופת מחקר הדוקטורט של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

Eldar Fischer and Eyal Rozenberg. Lower bounds for testing forbidden induced substructures in bipartite-graph-like combinatorial objects. In *Proceedings of RANDOM 2007*, pages 464–478. Springer, 2007.

Eldar Fischer and Eyal Rozenberg. Inflatable graph properties and natural property tests. In *Proceedings of RANDOM 2011*, pages 542–554, Berlin, Heidelberg, 2011. Springer-Verlag.

Oded Goldreich, Michael Krivelevich, Ilan Newman, and Eyal Rozenberg. Hierarchy theorems for property testing. In Oded Goldreich, editor, *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 289–294. Springer, 2010.

הכרת תודה מסורה לטכניון על מימון מחקר זה.

חסמים תחתונים ותוצאות מבניות בבדיקת תכונות של מבנים קומבינטוריים צפופים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

אייל רוזנברג

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
טבת התשע"ב חיפה ינואר 2012

חסמים תחתונים ותוצאות מבניות
בבדיקת תכונות
של מבנים קומבינטוריים צפופים

אייל רוזנברג