EVERY PROPERTY OF HYPERFINITE GRAPHS IS TESTABLE*

ILAN NEWMAN† AND CHRISTIAN SOHLER ‡

Abstract. A k-disc around a vertex v of a graph $G = (V, E)$ is the subgraph induced by all vertices of distance at most k from v. We show that the structure of a planar graph on n vertices, and with constant maximum degree d, is determined, up to the modification (insertion or deletion) of at most ϵdn edges, by the frequency of k-discs for certain $k = k(\epsilon, d)$ that is independent of the size of the graph. We can replace planar graphs by any hyperfinite class of graphs, which includes, for example, every graph class that does not contain a set of forbidden minors.

A pure combinatorial consequence of this result is that two d-bounded degree graphs that have similar frequency vectors (that is, the ℓ_1 difference between the frequency vectors is small) are close to be isomorphic (where close here means that by inserting / deleting not too many edges in one of them, it becomes isomorphic to the other).

We also obtain the following new results in the area of property testing, which are essentially equivalent to the above statement. We prove that

- graph isomorphism is testable for every class of hyperfinite graphs,
- every graph property is testable for every class of hyperfinite graphs,
- every hyperfinite graph property is testable in the bounded degree graph model,
- A large class of graph parameters is approximable for hyperfinite graphs.

Our results also give a partial explanation of the success of motifs in the analysis of complex networks.

Key words. Property Testing, graph properties

AMS subject classifications. F.2.2- Analysis of Algorithms and Problem Complexity. Non-numerical Algorithms and Problems
G.2.2- Discrete Mathematics, Graph Theory, Graph algorithms.

1. Introduction.

Given two planar graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ on n vertices whose maximum degree is bounded by a constant d, can we decide whether the two graphs are isomorphic? This problem is a special instance of the graph isomorphism problem, the complexity of which is not yet fully understood (but which has a polynomial algorithm for bounded degree graphs [14, 16]). Assume that we only want to solve a relaxed version of the problem, where we are supposed to accept the two graphs with probability at least 2/3 if they are isomorphic, and reject them if they have edit distance more than ϵdn. Further assume that we are given access to the adjacency list representation of the two graphs. This puts this question into the framework of property testing of bounded degree graphs, introduced in [11]. We show that with the promise that the graphs, say, are planar[1], the answer is positive in a very strong sense. Namely, there is an algorithm that for any ϵ, queries only a constant $q = q(\epsilon, d)$ number of vertices in the graphs and gets the (at most) d neighbors of every queried vertex. The algorithm accepts with probability 2/3 two planar graphs that are isomorphic while it rejects with probability 2/3 two planar graphs that are

*A preliminary version appeared in 43rd ACM Sypm. on Theory of Computing (STOC2011), June 6–8, 2011, San Jose, California.

† Department of Computer Science, University of Haifa, Haifa, Israel, email: ilan@cs.haifa.ac.il
This Research was supported by The Israel Science Foundation (grant number 862/10.)

‡ Department of Computer Science Technische Universität Dortmund, email: christian.sohler@tu-dortmund.de This research was supported by the German Research Foundation (DFG), grant So 514/3-2. The second author acknowledges the support of ERC grant No. 307696.

[1] The class of planar graphs here can be replaced with any hyperfinite class of graphs. The family of hyperfinite graphs contains bounded degree planar graphs, as well as any family that is defined by a collection of forbidden minors, and some other families of graphs.

ϵ-far from being isomorphic[2].

We also show that for *any* graph property Π, deciding whether a given planar graph has Π or is ϵ-far from having Π requires at most $q = q(\epsilon, d)$-queries - a constant number of queries to the graph (independent of the graph size). Combining this with a result of Benjamini, Schramm and Shapira [4] (or Hassidim, Kelner, Nguyen, and Onak [12]) this implies that a similar task can be performed for arbitrary graphs of bounded degree, if the studied property is planar, i.e. all graphs that have the property are planar. The result can be extended to any hyperfinite graph property. Previously, this was only known for monotone hyperfinite graph properties, which contain, for example, all minor-closed graph properties [4] (see also [12] for a simpler proof for the testability of all minor-closed graph properties).

Another immediate corollary of our results is that every *graph parameter*, i.e. every function of a graph that has the same value for isomorphic graphs, whose value changes by at most a constant under insertion or deletion of one edge, can be approximated up to an additive error of ϵdn for bounded degree planar graphs, using only a constant number of queries to the input graph. Again the result can be extended to any family of hyperfinite graphs. Previously, this was known only for specific functions like maximum matching, minimum vertex cover, minimum dominating set and maximum independent set [17, 12, 7] and for all parameters of graphs of bounded growth [8].

In the next couple of paragraphs we briefly describe the recent relevant developments in property testing of bounded degree graphs and what is the context of our contribution. As already noted in [4], our understanding of property testing in the dense graph model is much better than that of the bounded-degree model. This is since Szemerédi's Regularity Lemma provides a 'constant-size' description for any dense graph (up to changing $\epsilon|V(G)|^2$ of the edges / non-edges) that is accurate enough to test any testable graph property in that model [1]. For the bounded-degree model, we do not have such a theory. It turns out, however, that for bounded degree graphs that come from a hyperfinite-family of graphs, (e.g., planar graphs) the ideas from [6, 4, 12] together with the results here, provide a notion of 'constant-size' description of such graphs. This description is just the frequency vector of constant size discs in the graph. This notion serves us well from two points of view. It can be estimated very efficiently (by sampling a constant number of vertices and edges in the graph), and it captures enough information to let one decide the existence of every property of such graphs (up to the deletion or insertion of a small fraction of the edges). These two features are just what is needed for property-testing (in analogy with regular partitions as in [1]), and derive the corresponding results.

The first step in this direction was done by Czumaj, Shapira and Sohler in [6], where the authors showed that in planar graphs (again this can be extended to more general families of graphs) testing of hereditary graph properties can be reduced to testing the occurrence of small connected induced subgraphs. The break-through made by Benjamini, Schramm and Shapira [4], was to prove that this local view already says enough about the possibility of being planar (or not having any minor from a set of forbidden ones). They show that any graph which can be partitioned into small connected components by removing ϵdn edges, has a significantly different distribution of certain constant sized subgraphs than graphs, which are far from hav-

[2] ϵ-far here means that one needs to modify at least an ϵ-fraction of the edges in one of the graphs so that the resulting graphs become isomorphic. We will refer to such algorithms as *property testing algorithms* or *testers*, for short. See exact definitions in Section 2.

ing such a partition. Their proof shows that, based on the local information, one can (randomly) construct a global partition of the graph into small components by removing at most ϵdn edges from the graph. As a result they show that every monotone hyperfinite property of graphs is testable in constant time. This implies, in particular, that every minor-closed graph property is testable. In a follow-up work, Hassidim, Kelner, Nguyen, and Onak [12] give an explicit algorithm to locally compute such a partition. They develop a uniform tester for minor-closed properties and also improve on the running time of the previous tester. Furthermore, they show that the distance to any non-degenerate hereditary property can be approximated with additive error, if the graph comes from a hyperfinite family of graphs (a hereditary graph property is non-degenerate, if for any number of vertices the empty graph has the property). Their work uses a previous result by Nguyen and Onak, which shows how to transform certain greedy algorithms into local algorithms.

Using the machinery of [12] we show that this local view is sufficient for testing any property of planar graphs (or hyperfinite graphs in general). Our techniques heavily rely on the recent development in the area of testing in the bounded degree model described above. In particular, we use the local partition oracles used implicitly in [4], and explicitly in the later result by Hassidim, Kelner, Nguyen, and Onak [12], which in turn is based on a technique developed in [17].

Building on the results in the conference version of this paper, Elek extended our main result to the theory of graph limits and proved that if two hyperfinite graphings have the same local statistics then they are globally close [9].

We end this section by pointing at an interesting connection between the recent work in property testing and the concept of *motifs* used in the analysis of complex networks [15]. Motifs are subgraphs that occur significantly more frequent in a given class of graphs than in a random graph. They are supposed to be the 'building blocks' of networks appearing, for example, in biological applications and can also be used to classify certain classes of networks. Furthermore, they are often assumed to have a specific function in the considered class of networks. From a combinatorial point of view, a motif is simply a heavy hitter (large entry) in the histogram of constant sized subgraphs. A different view of our result says that two graphs are close to be isomorphic if their histograms of local neighborhoods are close. Thus, such heavy hitters should be of high significance to the structure of the whole graph, at least, if the graph is hyperfinite.

Finally, we remark that in a preliminary version of this paper in STOC2011, we have presented a different proof of the main results. We first proved the pure combinatorial result stated in Theorem 3.1 using a probabilistic argument. Then we showed that it directly implies the results on property testing. Here we present a much simpler direct proof for the property testing results, and show how the combinatorial statement follows. For those that are less interested in the theory of property testing we include also a direct proof of the purely combinatorial Theorem 3.1.

The rest of this draft is organized as follows: Section 2, contains basic notations, preliminary definitions and background on property testing. Section 3 contains the formal description of the results. Sections 4, and 5 describe the needed machinery from [12] along with an estimator for the frequency vector of local neighborhoods. Sections 6 and 7 contain the proofs of the main property testing results. Sections 8 and 9 contain the proofs related to isomorphism testing and the combinatorial results. Finally, we end with some additional discussion in Section 10.

2. Notations and definitions. In this paper we consider undirected *labeled* graphs without self-loops. We use $G = (V, E)$ to denote a graph with vertex set V and edge set E. We write $V(G)$ to denote the vertex set of graph G, and will always assume that $V(G) = \{1, \dots, n\}$ for the graph G at hand. We will shortly say that a graph is d-bounded degree if its maximum degree is at most d (here $d = O(1)$ will be a constant that does not depends on n, while the number of vertices n will tend to infinity).

To state our results we need the following definition of *hyperfinite* graphs, which was introduced in [7].

DEFINITION 2.1 (Hyperfinite). *Let $0 \le \epsilon \le 1$, and $k \in \mathbb{N}$. A graph G is called (ϵ, k)-hyperfinite, if one can remove ϵn edges from G and obtain a graph whose connected components have size at most k. For a function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ a graph G is called ρ-hyperfinite , if, for every $\epsilon > 0$, G is $(\epsilon, \rho(\epsilon))$-hyperfinite. A collection of graphs is ρ-hyperfinite, if every graph in the collection is ρ-hyperfinite. A collection of graphs is called hyperfinite, if there exists a function ρ such that the collection is ρ-hyperfinite.*

We note that d-bounded degree planar graphs are ρ-hyperfinite for $\rho = O(d^2/\epsilon^2)$, and any class that is defined by a finite collection of forbidden minors is ρ-hyperfinite for some ρ that is $O(d^2/\epsilon^2)$ (see e.g., [3]).

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *isomorphic* if there exists a bijective mapping $\Phi : V_1 \to V_2$ such that $(u, v) \in E_1$ if and only if $(\Phi(u), \Phi(v)) \in E_2$. Such a map Φ is called a *graph isomorphism map* between G_1 and G_2.

2.1. k-discs. A connected graph $G = (V, E)$ with a specially marked vertex v, is called a *rooted graph* and we sometimes say that G is rooted at v. A rooted graph $G = (V, E)$ with a root v has radius k if every vertex in V has distance at most k from v. Two rooted graphs G and H are isomorphic, if there is a graph isomorphism map between H and G that identifies the roots with each other. We denote by $N(k, d)$ the number of all non-isomorphic rooted graphs with maximum degree at most d and radius at most k. For a d-bounded degree graph $G = (V, E)$, an integer k and a vertex $v \in V$, let $B_G(v, k)$ be the subgraph rooted at v that is induced by all vertices of G that are at distance k or smaller from v. In particular, $B_G(v, k)$ is a graph of radius at most k. $B_G(v, k)$ will be called the k-disc around v. The k-discs of G are all possible $B_G(v, k)$, $v \in V$. Note that for bounded degree graphs, the number of all possible non-isomorphic k-discs is at most $N(k, d)$. Let $\mathcal{H}(k, d) = \{H_1, \dots, H_{N(k,d)}\}$ denote the set of all d-bounded degree rooted graphs with radius at most k.

2.2. Distance between graphs and Property testing. A graph property is a (possibly infinite) collection of graphs, which is closed under isomorphism.

DEFINITION 2.2 (Graph distance). *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be d-bounded degree graphs on n vertices. The distance $dist(G_1, G_2)$ is the number of edges that needs to be deleted and/or inserted from G_1 in order to make it isomorphic to G_2. We say that G_1, G_2 are ϵ-far from being isomorphic (or G_1 is ϵ-far from G_2), if $dist(G_1, G_2) > \epsilon dn$. Otherwise, we say that they are ϵ-close (to be isomorphic).*

DEFINITION 2.3 (ϵ-far). *Let Π be any (non-empty) graph property on d-bounded degree graphs. A d-bounded degree graph $G = (V, E)$ is said to be* ϵ-far *from Π, if it is ϵ-far from every $G' \in \Pi$. If G is not ϵ-far from Π, it is said to be* ϵ-close *to* Π.

A d-bounded degree graph will be represented by its adjacency lists. This can be thought of as an array of size $n \times d$ in which the ℓ-th row contains the names of the (at most) d neighbors of the ℓ-th vertex. This representation, in the context of property-testing, is referred to as the *bounded degree model* [11]. In this model, for a query to a vertex $v \in V$ and a number i, the i-th neighbor of v is returned in constant time. If v has less than i neighbors, a special symbol is returned to indicate this fact. For simplicity, as we ignore constants, we assume that a query in this model, for $d = O(1)$, is defined by a vertex $v \in V(G)$, and the result of a query is the set of the (at most d) vertices that are the neighbors of v.

The notion of property testing was introduced by Rubinfeld and Sudan [20] in the context of algebraic properties and then defined for combinatorial properties (in particular, for graph properties) by Goldreich, Goldwasser and Ron [10]. An ϵ-test for property Π, or for short "tester", in the bounded degree graph model, is an algorithm that is given as input a bounded degree graph G which it accesses via queries as described above. It accepts every graph from Π with probability at least $2/3$, and rejects every graph that is ϵ-far from Π with probability at least $2/3$.

If the graph neither has property Π nor is ϵ-far from Π, a property tester for Π may accept or reject. The query complexity of a tester is the number of queries for the worst case input and worst case random choices. We call a graph property testable, if it has a (non-uniform) property tester with constant query complexity.

Definition 2.4 (Testable graph properties). *A graph property* Π *is called* (non-uniformly) testable *in the bounded degree graph model with degree bound* d*, if there is a function* $q = q(\epsilon, d)$ *such that for any* $0 < \epsilon < 1$ *and any* $n \in \mathbb{N}$*, there is a tester* $A_{\epsilon,d,n}$ *for* Π*, whose query complexity on graphs of* n *vertices is* q*.*

A randomized algorithm is said to be *non-adaptive*, if its queries do not depend on answers of previous queries and thus a non-adaptive algorithm can specify all queries (or rather a probability distribution on sets of queries) right at the beginning of the algorithms. It is not hard to see that a constant time non-adaptive algorithm cannot do much in bounded-degree sparse graphs as the probability to sample two adjacent vertices tends to zero as the graph size increases. Hence, the only information one can obtain by sampling is an approximation of the distribution of the vertex degrees (see also [21] for a formal proof of this statement). We therefore formalize the notion of *weak non-adaptivity.* Consider an algorithm that specifies on its start, a (possibly random) set of vertices to be queried and for each vertex a sampling radius. Then the algorithm queries for each sampled vertex the subgraph induced by all vertices at distance that is at most its sampling radius. If these are the only queries made by the algorithm, it is called *weakly non-adaptive.* We note that, if an algorithm queries on every run all vertices in the discs of a certain predefined radius k around a uniformly chosen sequence of vertices of a fixed size then the algorithm is weakly non-adaptive. Namely, for some fixed integers k, r, the queries are to a random sequence $v_1, \ldots, v_r \in V(G)$ and the vertices in $B_G(v_i, k)$, $i = 1, \ldots, r$. We will use such an algorithm in Section 5.

All the algorithms that are considered in this paper are weakly non-adaptive.

2.3. Partitions and the local view of the graph. For a graph $G = (V, E)$, and a set $S \subseteq V$, we denote by $G[S]$ the induced graph on S, namely the graph (S, E') where $E' = \{(u, v) \in E \mid u, v \in S\}$. For two sets $A, B \subseteq V$ we denote $e(A, B) = |\{(u, v) \in E \mid u \in A,\ v \in B\}|$. A partition of a set V is a set of pairwise disjoint, non-empty subsets of V whose union is V. For a partition $\mathcal{P} = \{C_1, \ldots, C_r\}$ of $V(G)$ we denote by $G[\mathcal{P}]$ the graph that is the union of $G[C_i]$. Note that $G[\mathcal{P}]$ is disconnected if $r \geq 2$ and is obtained from G by deleting all edges whose end points are in different partition classes.

In this paper we will show that, for some $k = O(1)$, knowing the number (or even an approximation of it) of each type of k-disc in a graph taken from a hyperfinite-family of graphs already determines, from the point of view of property testing, the presence of any graph property for that graph. To make this formal we use the following definition.

Definition 2.5 (Local view of a graph). *For a d-bounded degree graph $G = (V, E)$ and integer k, let $hist_G(k)$ be the histogram vector of all k-discs of G. Namely, $hist_G(k)$ is a vector of dimension $N(k, d)$, indexed by all possible rooted graphs of radius at most k and degree at most d. The ith entry of $hist_G(k)$ corresponds to $H_i \in \mathcal{H}(k, d)$, and counts the number of k-discs of G that are isomorphic to H_i. Note that G has $n = |V|$ different discs, thus the sum of entries in $hist_G(k)$ is n. Let $freq_G(k)$ be the normalized distribution, namely $freq_G(k) = hist_G(k)/n$.*

For a vector $v = (v_1, \ldots, v_r)$ we will use $\|v\|_1 = \sum_{i=1}^{r} |v_i|$ to denote its ℓ_1-norm. We say that two unit-length vectors u, v are λ-close, if $\|u - v\|_1 \leq \lambda$.

We present here the following two basic claims that relate the distance between frequency vectors of graphs and the distance between the graphs.

Claim 2.1. *Let G_1, G_2 be two d-bounded degree graphs on n vertices. Let $k \geq 0$ be an integer and $0 < \lambda < 1$. If $freq_{G_1}(k)$ and $freq_{G_2}(k)$ are λ-close then there is a bijection $\Psi : V_1 \mapsto V_2$ such that, for all $v \in V_1$ but a λ-fraction of the vertices, $B_{G_1}(v, k)$ is isomorphic to $B_{G_2}(\Psi(v), k)$.*

Proof. Let $\mathcal{H}(k, d) = \{H_1, \ldots, H_{N(k,d)}\}$ be as defined above. For every $i \in [N(k, d)]$, let $\alpha_1(H_i)(\alpha_2(H_i))$ denote the number of occurrences of H_i as a rooted graph in G_1 (G_2 respectively). Since $|V_1| = |V_2| = n$, the assumption that $\|freq_{G_1}(k) - freq_{G_2}(k)\|_1 \leq \lambda$ implies that $\|hist_{G_1}(k) - hist_{G_2}(k)\|_1 = \sum_{i \in [N(k,d)]} |\alpha_1(H_i) - \alpha_2(H_i)| \leq \lambda \cdot n$.

For each $i \in [N(k, d)]$, we can pair greedily the vertices $v \in V_1$ with $u \in V_2$ if both $B_{G_1}(v, k), B_{G_2}(u, k)$ are isomorphic to H_i. This will leave out at most $|\alpha_1(H_i) - \alpha_2(H_i)|$ vertices unpaired. This pairing is extended to a bijection Ψ between V_1 and V_2 arbitrarily on unpaired vertices. This leaves out altogether $\sum_{i \in N[k,d]} |\alpha_1(H_i) - \alpha_2(H_i)| \leq \lambda n$ vertices $v \in V_1$ for which $B_{G_1}(v, k)$ is not isomorphic to $B_{G_2}(\Psi(v), k)$. ∎

Claim 2.2. *Let $k \geq 1$ be an integer and let $0 < \lambda < 1$. Let G_1^* and G_2^* be two d-bounded degree graphs on n vertices whose connected components have size at most k and whose frequency vectors $freq_{G_1^*}(k)$ and $freq_{G_2^*}(k)$ are λ-close. Then G_1^* and G_2^* are λ-close.*

Proof. Greedily map connected components of G_1^* to isomorphic ones in G_2^*, until this is no longer possible, and then extend the map to a bijection between the remaining subsets of vertices arbitrarily. By definition, only a λ-fraction of the vertices will be mapped to images on which the k-neighborhoods (which are equivalent to the connected component of the vertex) do not agree. Deleting the incident edges from G_1^* and inserting the corresponding edges from G_2^* under our bijection requires at most λdn edge modifications by the bounded degree assumption. Hence, the graphs are λ-close. ∎

3. Results. Our main combinatorial result is the following theorem. It states that the local view of a hyperfinite graph determines, up to changing at most ϵdn edges, its global structure.

Theorem 3.1 (Local versus global graph structure). *Let G_1, G_2 be d-bounded degree, ρ-hyperfinite graphs on n vertices. Then for every ϵ, $0 < \epsilon \leq 1$, there exists $\eta = \eta(\epsilon, \rho, d)$, $D = D(\epsilon, \rho, d)$, such that, if $\|freq_{G_1}(D) - freq_{G_2}(D)\|_1 \leq \eta$ then G_1 is ϵ-close to being isomorphic to G_2.*

In the following, we will derive a number of results in the area of property testing.

These results are (essentially) equivalent to Theorem 3.1. The first result says that graph isomorphism is testable for hyperfinite graphs. This follows almost immediately from our main theorem (and also vice versa, if one also takes the structure of the property tester into account). The reason is that, in the context of property testing, the local view of a graph gives a sufficiently close estimation of its global view. In addition, one can estimate the local view by random sampling. The other results are related to Theorem 3.1 in a similar way.

One may consider Theorem 3.1 as our main result, as it is purely combinatorial and meaningful outside the area of property-testing. Indeed in a first version [18] we prove it first and deduce the other implications. However, as it turns out, proving directly Theorem 3.3 is conceptually easier. Hence we delay the proof of Theorem 3.1 to the end.

THEOREM 3.2 (Isomorphism of hyperfinite graphs is testable). *Let $\mathcal{C}$ be a ρ-hyperfinite family of graphs with maximum degree at most d. Then there is a function $s = s(\epsilon, \rho, d)$ such that for any $0 < \epsilon < 1$ there is a tester $A_{\epsilon,d}$ with query complexity s for the graph isomorphism problem for graphs in $\mathcal{C}$. Namely, the algorithm $A_{\epsilon,d}$ has access to two graphs $G_1, G_2 \in \mathcal{C}$ with $|V_1| = |V_2|$. It makes at most s queries to G_1 and G_2 and accepts G_1, G_2 with probability at least $2/3$, if G_1 is isomorphic to G_2, and rejects with probability at least $2/3$ if G_1 is ϵ-far from any isomorphic copy of G_2. If the function ρ is computable, then there is also a uniform tester for graph isomorphism.*

Following a similar line of thought one can also prove that any graph property is testable in hyperfinite graphs.

THEOREM 3.3 (Every property is testable for hyperfinite graphs). *Let $\mathcal{C}$ be a hyperfinite family of d-degree bounded graphs, and Π any graph property. The property Π is testable for any graph taken from $\mathcal{C}$.*

Note that Theorem 3.3 is a 'conditional' statement. Namely, if the graph G is in the hyperfinite family $\mathcal{C}$ then it can be tested for Π. To obtain the following unconditional theorem we use the fact that being in a hyperfinite family $\mathcal{C}$ can be tested, similarly to [4]. Together with the previous theorem, this implies that we can test every property that is the intersection of a hyperfinite family $\mathcal{C}$ (which is closed under isomorphism) and an arbitrary graph property. Equivalently, every graph property which contains only hyperfinite graphs can be tested. Such a graph property will be called hyperfinite.

THEOREM 3.4 (Every hyperfinite property is testable). *Every hyperfinite graph property is testable in the bounded degree graph model.* In particular, taking Π to be the set of all graphs, the above implies the testability of being in $\mathcal{C}$, e.g., rephrases that planarity is testable (by the same test as suggested in [12]), which was first shown in [4].

An immediate application of Theorem 3.3 is that of approximating graph parameters. A graph parameter is an integer function that maps graphs to a range $\{1, \ldots, m\}$ and that assigns the same value to isomorphic graphs. Examples include the size of a maximum matching, a maximum cut or independent set. We say that a graph parameter f is Δ-robust, if, for any G, $f(G)$ changes by at most $\pm\Delta$, if an edge is added to or deleted from G. Let f be a Δ-robust graph parameter for $\Delta = O(1)$. Let $\Pi(\ell, \epsilon)$ be the set of all graphs G for which $\ell - \epsilon d|V| \leq f(G) \leq \ell$. Since $\Pi(\ell, \epsilon)$ is a graph property, Theorem 3.3 asserts that $\Pi(\ell, \epsilon)$ can be tested using a constant number of queries for any hyperfinite family of graphs. Thus an estimate of $f(G)$ can be approximated to within an additive error of $\epsilon d|V|$ by testing $\Pi(\ell, \epsilon)$ for various $\ell's$.

Thus we get,

COROLLARY 3.5. *Let $\mathcal{C}$ be a ρ-hyperfinite family of graphs of d-bounded degree, and f be any $O(1)$-robust graph parameter. Then for any $0 < \epsilon \leq 1$ there is a constant query complexity, randomized algorithm $A_{\epsilon,d,n}$, that approximates $f(G)$ to within an additive error of $\epsilon d|V|$, for any graph $G = (V, E) \in \mathcal{C}$, $|V| = n$, with probability at least $2/3$.*

4. Algorithms for local graph partitioning. We present here the main machinery from [12]. Our proof requires their construction of a partitioning oracle (see Definition 4.2) as stated in Lemma 4.3 below. Every graph taken from a hyperfinite family of graphs admits a partitioning into small connected components by removing a fraction of the edges. To be useful for property testing and sublinear approximation algorithms, it would be nice if the features of such partitions could be obtained by some local sampling. Indeed an oracle to such a partition, that decides in constant time, for each vertex, to which partition class it belongs, has been developed by Hassidim, Kelner, Nguyen, and Onak [12]. Also, an earlier work of Benjamini, Schramm and Shapira [4] implicitly contains a way to construct such a local partitioning. In the following we will explain the result of Hassidim et al. in more details as their algorithm will be required for our analysis.

The oracle is constructed by using a technique from [17] that can be applied to derive constant time algorithms from certain greedy approximation algorithms. In [12] the authors present a greedy algorithm to compute the desired partition, which then can be made local using the technique from [17]. In this greedy algorithm, the vertices are considered in random order $\pi = (\pi_1, \ldots, \pi_n)$. The algorithm greedily removes components containing the current vertex π_i (if this vertex is still in the graph when it is considered by the algorithm). Namely, if there exists a small connected set S of vertices that contains π_i and that has a small cut to the rest of the graph, this set will be cut out by the algorithm. Otherwise the single vertex π_i is cut out.

Formally, a set $S \subseteq V$ is a (k, ξ)-*isolated neighborhood* [12] of $v \in V$, if $v \in S$, the subgraph induced by S is connected, $|S| \leq k$ and $e(S, V - S) \leq \xi|S|$. With this definition, we can state the algorithm from [12] that computes a partition of G.

GLOBALPARTITIONING(k, ξ)
 $\pi = (\pi_1, \ldots, \pi_n)$ = random permutation of $V(G)$.
 $\mathcal{P} = \emptyset$,
 while G is not empty **do**
  Let v be the first vertex in G according to π
  if there exists a (k, ξ)-isolated neighborhood of v in G
   then S = this neighborhood
  else $S = \{v\}$
  $\mathcal{P} = \mathcal{P} \cup \{S\}$
  Remove all vertices in S from the graph

Note that the random permutation of the vertices deterministically defines the partition. We thus denote this partition by $\mathcal{P}(\pi)$ and the graph induced by the connected components of $\mathcal{P}(\pi)$, by $G[\mathcal{P}(\pi)]$. The partition, of course, depends also on ξ and k, but these parameters will be taken to be fixed in the context of use. For any run, namely choice of π, the connected components in $G[\mathcal{P}(\pi)]$ are of size at most k. If in addition at most ϵn edges are deleted in order to move from G to $G[\mathcal{P}(\pi)]$, we say that the partition is an (ϵ, k)-partition. Hassidim et al. [12] proved.

THEOREM 4.1. [12] *Let $0 < \epsilon < 1$ and G be a d-bounded degree graph taken from* *a ρ-hyperfinite family. Then there is a $\xi = \xi(\epsilon, \rho, d)$, $k = k(\epsilon, \rho, d)$ such that, with probability at least $9/10$, the random partition generated by the randomized algorithm above for G, is an $(\epsilon d, k)$-partition.*

If we are to make only few queries to the graph, we need a more local view of such partitions. This motivates the following definition of [12]. Recall that a query of a vertex $v \in V(G)$ returns all the neighbors of v. In the following definition $\mathcal{P}[v]$ denotes the set of vertices of the partition that contains v.

Definition 4.2. [12] *A randomized algorithm $\mathcal{O}$ is an (ϵ, k)*-partitioning oracle *for a class $\mathcal{C}$ of graphs, if, given query access to a graph $G = (V, E)$, it provides query access to a partition $\mathcal{P}$ of V. For a query about $v \in V$, $\mathcal{O}$ returns $\mathcal{P}[v]$; the partition class that contains v. The partition has the following properties:*

- *$\mathcal{P}$ is a function of the graph and random bits used by the oracle. In particular, it does not depend on the order of queries to $\mathcal{O}$.*
- *For every $v \in V$, $|\mathcal{P}[v]| \leq k$ and $\mathcal{P}[v]$ induces a connected graph in G.*
- *If G belongs to $\mathcal{C}$, then $|\{(v, w) \in E : \mathcal{P}[v] \neq \mathcal{P}[w]\}| \leq \epsilon |V|$ with probability $9/10$.*

For sake of completeness, we will briefly recap the ideas from [12] how to turn the global partitioning algorithm in a local partitioning oracle. In [12] it is shown how to implement the global partition algorithm described above such that one can access the global partition using only queries to a small local part of the graph. The result of this implementation is a (local) partitioning oracle. In order to do so, one issue is to simulate locally a random permutation. This is done as follows. Each vertex is assigned a random priority from $[0, 1]$, independently of any other vertex. Thus this defines a random map $\pi : V \mapsto [0, 1]$. Such a map naturally defines a permutation on V by taking the vertices according to increasing priorities[3]. The advantage of defining a permutation by such random ordering is that one can select the priority at random whenever access to a vertex (and its priority) is required. In what follows we will identify such a map π with the random permutation on V that is associated with it.

Now, a local algorithm can access the partition $\mathcal{P}[v]$ of a query vertex v as follows: The $2k$-disc around v is explored, and whenever a new vertex is encountered whose priority has not yet been fixed, its priority is chosen uniformly at random from $[0, 1]$. If v has lowest priority among all discovered vertices in the neighborhood, we know that it would be encountered and cut out first in its k-disc by the algorithm GlobalPartitioning running on a permutation that is consistent with the (partially determined) random order π. Hence, this operation cannot affect vertices that have been cut out earlier in other places of the graph. Indeed, any vertex u with lower priority must have a distance of more than $2k$ from v and cutting out vertices never decreases distances, so any k-neighborhood of u cannot intersect any k-neighborhood of v. Thus, in this case, the partition class $\mathcal{P}[v]$ is determined and can be answered by the oracle. Otherwise, if the neighborhood around v contains a vertex u with $\pi(u) < \pi(v)$, then we recurse by moving to the vertex u with the smallest priority. We note that the oracle can be used to answer a sequence of queries, in such a case, the partially determined π is, of course, saved. Namely, once a vertex is encountered along the above process, its priority is determined and will never change in the future.

We also note that for every permutation π the local partition oracle is consistent with the partition produced by the algorithm GlobalPartitioning above, for the same π (and the same ϵ and k). It is proved in [12] that the expected number of queries

[3] If two vertices are mapped to the same value, then the permutation is not well defined, but this occurs with probability 0, hence will be disregarded.

(namely the recursion depth) is bounded for hyperfinite graphs. This is summed up in the following lemma, which is a slight variation of the lemma stated in [12] (see also [19]).

LEMMA 4.3. [12] *Let G be an $(\epsilon^3/54000, k)$-hyperfinite d-bounded degree graph with $d \geq 2$. Then there is an $(\epsilon d, k)$-partitioning oracle with the following properties: If the oracle is asked q non-adaptive queries, then with probability $1-\delta$, the oracle makes $\frac{q}{\delta} \cdot 2^{d^{O(k)}}$ queries to the input graph. The time complexity for computing the answers to the q queries is bounded by $\frac{q}{\delta} \log \frac{q}{\delta} \cdot 2^{d^{O(k)}}$.*

5. Estimating the frequency vector of a graph. For our testers, as well as for the proof, we need to approximate the frequencies of D-discs in a graph $G=(V,E)$. The vector $freq_G(D)$ can be estimated to within additive error $\pm\lambda$ by sampling a constant number of vertices and exploring the D-discs around them. We give the algorithm below.

EstimateFrequencies($G=(V,E)$ with maximum degree d, D, s)
  $\widetilde{freq}_G(D) = 0$
    sample s vertices $u_1, \ldots, u_s$ uniformly at random
    for $j = 1$ to s **do**
      explore the D-disc around u_j
      let i be the index such that $H_i \in \mathcal{H}(D,d)$ is isomorphic to this D-disc
      set $\widetilde{freq}_G(D)[i] = \widetilde{freq}_G(D)[i] + \frac{1}{s}$
  return $\widetilde{freq}_G(D)$

LEMMA 5.1. *Let G be a d-bounded degree graph. If $s \geq \frac{N(D,d)^2}{\lambda^2} \cdot \ln(N(D,d)+40)$ then algorithm* EstimateFrequencies *above returns a vector $\widetilde{freq}_G(D)$ that with probability at least $19/20$ satisfies*

$$\|\widetilde{freq}_G(D) - freq_G(D)\|_1 \leq \lambda.$$

Proof. Let $X_{i,j}$ be the indicator random variable for the event that the D-disc $B_G(u_j, D)$ is isomorphic to H_i. We observe that this event happens with probability $freq_G(D)[i]$. This implies that $\mathbf{E}[X_{i,j}] = freq_G(D)[i]$ and $\mathbf{E}[\sum_{j=1}^{s} X_{i,j}] = s \cdot freq_G(D)[i]$. Furthermore, we have $\mathbf{E}[\widetilde{freq}_G(D)[i]] = \frac{1}{s} \cdot \mathbf{E}[\sum_{j=1}^{s} X_{i,j}] = freq_G(D)[i]$.

Using the Chernoff bound (see e.g., Theorem A.1.4 in [2]) we get for every i,

$$\begin{aligned}
&\mathbf{Pr}[|\widetilde{freq}_G(D)[i] - freq_G(D)[i]| > \lambda/N(D,d)] \\
&= \mathbf{Pr}[|\sum_{j=1}^{s} X_{i,j} - \mathbf{E}[\sum_{j=1}^{s} X_{i,j}]| > \frac{\lambda}{N(D,d)} \cdot s] \\
&\leq 2e^{-2\frac{\lambda^2 \cdot s}{N(D,d)^2}} \\
&\leq \frac{1}{20N(D,d)}
\end{aligned}$$

By the union bound we get that with probability at least $19/20$ for every i, $\|\widetilde{freq}_G(D)[i] - freq_G(D)[i]\|_1 \leq \lambda/N(D,d)$. This implies that $\|\widetilde{freq}_G(D) - freq_G(D)\|_1 \leq \lambda$. □

We end up this section with the following note: In the same way that the algorithm EstimateFrequencies can estimate the frequency vector of G, it can be used as well to estimate the frequency vector of $G[\mathcal{P}(\pi)]$, namely the graph that is induced by the $(\epsilon d, k)$-partition that is defined by an $(\epsilon d, k)$-partitioning oracle with random permutation π. One can just apply it on $G[\mathcal{P}]$ with $D = k$, and for each query to a vertex u and the exploration of the k-disc around it, one just runs the partition oracle that returns the connected component of u in $G[\mathcal{P}]$, as asserted by Lemma 4.3. Thus, each query in algorithm FrequencyEstimate is executed by calling the partition oracle. Note that algorithm EstimateFrequencies samples the vertices $u_1, \ldots, u_s$ non-adaptively and so the guarantees of Lemma 4.3 hold. We query the partitioning oracle for the partition classes of $u_1, \ldots, u_s$. These partition classes are the k-discs around $u_1, \ldots, u_s$ since every connected component in $G[\mathcal{P}]$ has diameter at most k.

This is formally stated in the following Lemma.

Lemma 5.2. *For every choice of constants δ, λ with $0 < \delta, \lambda < 1$ and every $k \geq 1$, there are values $\epsilon_{5.2} = \epsilon_{5.2}(\delta)$, $D_{5.2} = D_{5.2}(\lambda, k, d)$, $s_{5.2} = s_{5.2}(\lambda, k, d)$ and a randomized algorithm* Sampler, *that on a random permutation π (given as a priority vector), accesses a graph G with maximum degree at most d by querying independently $s_{5.2}$ random vertices of G and exploring the $D_{5.2}$-discs around them. The algorithm outputs a frequency vector $\tilde{f}$. If the input graph G is d-bounded degree and $(\epsilon_{5.2}, k)$-hyperfinite then, with probability at least $4/5$ (over the choices of π and the internal coins of the algorithm), the following two events occur.*

1. *The partition $\mathcal{P}(\pi)$ defined by algorithm* GlobalPartitioning *with parameters $\epsilon_{5.2}$, k, and using the permutation π as its random permutation, is a $(\delta d, k)$-partition.*
2. *The output vector $\widetilde{f}$ λ-approximates the frequency vector $freq_{G[\mathcal{P}(\pi)]}$ of the k-discs in $G[\mathcal{P}(\pi)](k)$. Namely, we have $\|\tilde{f} - freq_{G[\mathcal{P}(\pi)]}(k)\|_1 \leq \lambda$.*

Proof. Let δ, λ and k be given. Define $\epsilon_{5.2}(\delta) = \delta^3/54000$, the values required to apply Lemma 4.3. Then the definition of partitioning oracles implies that with probability at least 9/10 the first event occurs. Next, define $s_{5.2}(\lambda, k, d) = \frac{N(k,d)^2}{\lambda^2} \cdot \ln(20N(k, d))$, the value required to apply Lemma 5.1. Provided the oracle can answer all queries based on the information obtained from the sampled $D_{5.2}$-discs, this implies that with probability 19/20 the second event occurs. Finally, Lemma 4.3 and the fact that the local partitioning oracle accesses connected sets around each query vertex imply that, for sufficiently large $D_{5.2}(\lambda, k, d)$, with probability at least 19/20 all queries to the oracle can be answered within the queried discs. This proves our lemma. □

6. Every property is testable for hyperfinite graphs. We now have the machinery required for proving our results. We first start with the proof of Theorem 3.3. We restate the theorem below.

Theorem 6.1 (Every property is testable for hyperfinite graphs; restated). *Let $\mathcal{C}$ be a hyperfinite family of d-bounded degree graphs, and Π any graph property. The property Π is testable for any graph taken from $\mathcal{C}$.*

The main idea is quite simple. Let $G \in \mathcal{C}$, where $\mathcal{C}$ is ρ-hyperfinite and G is d-bounded degree. By the definition, since G is (ϵ, k)-hyperfinite for any ϵ and $k = \rho(\epsilon)$, G has a 'small description' up to the deletion of ϵdn edges. This is just the frequency vector $freq_{G[\mathcal{P}(\pi)]}(k)$ of the components of $G[\mathcal{P}(\pi)]$, where $\mathcal{P}(\pi)$ is an $(\epsilon \cdot d, k)$-partition of G. If $\mathcal{P}(\pi)$ is indeed an $(\epsilon \cdot d, k)$-partition of G, these components are all spanned by k-discs, and $G[\mathcal{P}(\pi)]$ is ϵ-close to G. Note that $freq_{G[\mathcal{P}(\pi)]}(k)$ is an $N(k, d)$ dimensional vector, that is $O(1)$-dimensional. The entries in $freq_{G[\mathcal{P}(\pi)]}(k)$ might not be of constant size (namely, have denominators n), but we avoid this issue at this stage (this will be automatically taken care of in the formal proof and test below). Furthermore, knowing $freq_{G[\mathcal{P}(\pi)]}(k)$ exactly, uniquely defines $G[\mathcal{P}(\pi)]$ - it is just being composed of the appropriate number[4] of disjoint copies of $H_i,\ i = 1, \dots, N(k, d)$.

Now assume that we want to ϵ-test G for a property Π, and that we have at hand the 'small description' $freq_{G[\mathcal{P}(\pi)]}(k')$ of G for some value $k' = k'(\epsilon, \rho)$ that is accurate up to $\epsilon dn/2$ edges. The test is then obvious: we reconstruct $G^* = G[\mathcal{P}(\pi)]$ and check whether it is $\epsilon/2$-close to Π. If it is we accept G and otherwise reject it. Indeed, if G has Π, then since G^* is $\epsilon/2$-close to G, G^* is $\epsilon/2$-close to Π and hence G will be accepted. On the other hand, if G is ϵ-far from Π, then G^* is $\epsilon/2$-far from Π and so the algorithm rejects.

Thus all we need is an access to the corresponding small description $freq_{G[\mathcal{P}(\pi)]}(k')$. While we can't have this, the sampler guaranteed by Lemma 5.2 gives an approximation to this small description. Thus all we need to do is to handle the accumulation of sampling errors, and the probability of failure. This is done formally below.

Proof. [Of Theorem 3.3] Let $G \in \mathcal{C}$, where $\mathcal{C}$ is ρ-hyperfinite and G is d-bounded degree. Let Π be the property that we want to ϵ-test G for. Let $\delta = \lambda = \epsilon/4$, $\epsilon' = \epsilon_{5.2}(\delta)$ and let $k = \rho(\epsilon')$, $D = D_{5.2}(\lambda, k, d)$. By Lemma 5.2 there is a randomized algorithm that makes at most $q = s_{5.2}(\lambda, k, d) \cdot d^{D+1} = O(1)$ queries to G and outputs a frequency vector $\tilde{f}$, with the guarantees of Lemma 5.2, that corresponds to δ and k.

This sampler defines the ϵ-test for Π (namely, all the queries it performs). The decision is then taken as follows. We check if there is a graph $G' \in \Pi \cap \mathcal{C}$ so that G' has a $(\delta d, k)$-partition $\mathcal{P}'$ and a corresponding induced graph $G'[\mathcal{P}']$, for which the frequency vector of k-discs, $freq_{G'[\mathcal{P}']}(k)$, is δ-close to $\tilde{f}$. If there is such G' we accept G and otherwise reject it. The query complexity of the test is obviously $O(1)$. We will now prove its correctness.

Disregarding whether G has Π or not, since $G \in \mathcal{C}$, and $\mathcal{C}$ is ρ-hyperfinite, it follows that G is (ϵ', k)-hyperfinite. Hence, Lemma 5.2 guarantees that with probability $4/5$ the sampler outputs a vector $\tilde{f}$ that is λ-close to a frequency vector of k-discs, $freq_{G^*}(k)$, of a graph $G^* = G[\mathcal{P}(\pi)]$, where $\mathcal{P}(\pi)$ is a $(\delta d, k)$-partition of G.

We assume in the following that this event happens, and prove then that the decision of the algorithm is correct. Indeed, assume that the event above, guaranteed by Lemma 5.2 occurs, and G has Π. Consider G itself with a partition $\mathcal{P}(\pi)$ for which $G[\mathcal{P}(\pi)]$ is a $(\delta d, k)$-partition, as a possible G' in the definition of acceptance. Obviously, for this G' the frequency vector $freq_{G'[\mathcal{P}']}(k)$ is identical to $freq_{G^*}(k)$. Recall however, that $\tilde{f}$ is δ-close to $freq_{G'[\mathcal{P}']}(k)$, hence G will be accepted.

Next we prove that, if the guarantees of Lemma 5.2 hold (which happens with probability at least $4/5$), then no graph that is ϵ-far from Π will be accepted. Let us assume that a graph G is accepted on account of a graph $G' \in \Pi \cap \mathcal{C}$ and a corresponding frequency vector $freq_{G'[\mathcal{P}']}(k)$ of $G'[\mathcal{P}']$, and that the guarantees of Lemma 5.2 hold. Then $\|\tilde{f} - freq_{G'[\mathcal{P}']}(k)\|_1 \le \delta$ (due to acceptance) and $\|\tilde{f} - freq_{G^*}(k)\|_1 \le \lambda$ (due to the guarantee of Lemma 5.2). Hence by the triangle inequality, $freq_{G^*}(k)$ is $(\delta + \lambda)$-close to $freq_{G'[\mathcal{P}']}(k)$. In turn, Claim 2.2 implies that G^* is $(\delta + \lambda)$-close to $G'[\mathcal{P}']$. This, in turn, implies that G is $(3\delta + \lambda)$-close to G'. Hence G is ϵ-close to Π. Thus, if G is ϵ-far from Π, it can only be accepted, if the guarantees of Lemma 5.2 do not hold, which proves our theorem. □

Remarks: We end this discussion with a few notes on the proof.

[4]Each H_i is counted $|V(H_i)|$ times, but not all of them are isomorphic as rooted discs. However, each fixed H_i contributes a fixed number to each entry of $hist_{G[\mathcal{P}(\pi)]}(k)[i]$.

- The assumption in the theorem that $\mathcal{C}$ is ρ-hyperfinite was only to assure that G is (ϵ', k)-hyperfinite for the ϵ' as in the proof. Hence, this assumption can be replaced with the weaker assumption "G being (ϵ', k)-hyperfinite for an appropriate ϵ' ". The same phenomenon occurs for all the theorems we present.
- Note that the test is somewhat oblivious of Π. Namely, for every Π it would make the same queries. Only the decision is dependent of Π. In particular, the test can be run without knowing Π, or for several $\Pi's$, making the same set of queries, and then making a decision for any desirable property (or all properties). This is also true for all the relevant theorems that we present.
- While the query complexity is fixed and independent of Π, we don't have a reasonable bounds on the time complexity (neither one can expect anything for this generality of statement, as Π may not even be decidable). Fixing the size of the graphs n, and fixing Π defines a finite collection $\mathcal{G}_n$ of graphs of n vertices - these in $\Pi \cap \mathcal{C}$. One could run the test by constructing all the frequency vectors for graphs in $\mathcal{G}_n$. Alternatively, one can choose the error parameters somewhat smaller, and construct a fine enough net for this collection of frequency vectors.

7. Every hyperfinite property is testable. Again we start by restating the theorem we are about to prove.

THEOREM 7.1 (Every hyperfinite property is testable). *Every hyperfinite graph property is testable in the bounded degree graph model.*

Proof. Let Π be a ρ-hyperfinite graph property, i.e. Π viewed as a family of graphs is ρ-hyperfinite. To ϵ-test that a given graph is in $\Pi \cap \mathcal{C}$ is very similar to the test proposed in the proof of Theorem 3.3. The difference is that we now do not have the promise that G comes from $\mathcal{C}$ and hence is not necessarily ρ-hyperfinite. The idea (which has first been developed in [4]) is very simple: we first "test" if G has $(\delta d, k)$-partition for a small enough δ. If it fails to have such a partition, it will be far from being ρ-hyperfinite, and hence we may safely reject. If it does have such a partition, then we apply the test in the proof of Theorem 3.3. As explained in the discussion following that proof, this premise is just enough to carry on the proof.

A tester that distinguishes (ϵ, k)-hyperfinite graphs from graphs that are not $(4\epsilon \log(4d/\epsilon), k)$-hyperfinite has been given in [4]. Here, we will use a simple test that has been used in [12] to obtain improved bounds for testing minor-closed properties.

To test whether the graph has $(\delta d, k)$-partition, we just run the local partition oracle guaranteed by Lemma 4.3 for some q random edges $(u, v) \in E(G)$ and estimate the fraction of edges cut by the partition (namely for which $\mathcal{P}(u) \neq \mathcal{P}(v)$). This obviously gives a good estimate for the number of edges cut by the partition. Formally: Let $\delta = \epsilon/4,\ \epsilon' = \epsilon_{5.2}(\delta),\ \delta_1 = \frac{\epsilon'}{4d},\ q = 1/\delta_1^2$ and $k = \rho(\delta_1^3/54000)$.

The ϵ-test for $\Pi \cap \mathcal{C}$ is done in two phases:

Phase 1: verifying that G is ρ-hyperfinite: We first perform q 'random edge' queries to the graph. Each query $Q_i,\ i = 1, \dots, q$ is implemented by selecting a vertex $v_i \in V(G)$ uniformly at random, and a random number $r_i \in [d]$. We make $Q_i,\ i = 1, \dots q$ independent. As a result we get a sequence of vertices $(v_i, \alpha_i),\ i = 1, \dots, q$ where α_i is either 'null' if $deg(v_i) < r_i$, or the r_ith neighbor of v_i, denoted u_i. Now, for each query Q_i for which we have obtained an edge (v_i, u_i), namely for which an r_ith neighbor of v_i exists, we find out whether the edge (v_i, u_i) is cut by the partition defined by the partition oracle. To do that we apply the random local partition oracle that is guaranteed by Lemma 4.3, with input parameter δ_1, to make the q nonadaptive vertex queries $\{v_i| \ i \in [q]\}$. Since the partitioning oracle returns the component $\mathcal{P}[v_i]$ of vertex v_i, we can check whether u_i is in this component or not, i.e. whether the edge (v_i, u_i) is cut. If v_i and u_i are in different components, we mark the query.

We accept this phase if at most $4\delta_1 \cdot q$ of the queries are marked. In that case we go to Phase 2. Otherwise, we reject Phase 1, and the whole test.

Phase 2: We perform the ϵ-test suggested in the proof of Theorem 3.3 for the property $\Pi \cap \mathcal{C}$.

Obviously the test has $O(1)$ query complexity. We prove below its correctness.

We first note that for any fixed edge $e = (v, u)$ the probability that this edge will be picked by a random query Q_i as above, (with either $v_i = v$ or $v_i = u$) is exactly $\frac{2}{dn}$. In particular, every edge has the same chance of appearing in a query.

Claim 7.1. *If $G \in \mathcal{C}$, then it passes Phase 1 with probability at least* 0.875.

Proof. Indeed, assume that $G \in \mathcal{C}$, then by the definition of $\mathcal{C}$ being ρ-hyperfinite, G is $(\delta_1^3/54000, k)$-hyperfinite. But then, by Lemma 4.3, with probability 9/10 the partition $\mathcal{P}$ that the oracle produces is consistent with a $(\delta_1 d, k)$-partition. Namely, at most $\delta_1 dn$ edges are cut by $\mathcal{P}$. Hence, for a random query Q_i, the probability that it is marked is at most $\delta_1 dn \cdot \frac{2}{dn} = 2\delta_1$. Given that the partition oracle is indeed as guaranteed in Lemma 4.3, we expect at most $2\delta_1 q$ queries to be marked, and by Chernoff (Theorem A.1.4. in [2]), the probability that more than $4\delta_1 q$ queries are marked is at most $e^{-8\delta_1^2 q} < 1/40$ for our choice of parameters. Thus such G will pass this stage with probability as claimed. □

The partial inverse (just what is needed for the tester of Theorem 3.3) is also true.

Claim 7.2. *If G is not (ϵ', k)-hyperfinite, then it will be rejected by Phase 1 with probability at least* 19/20.

Proof. Assume that G is not (ϵ', k)-hyperfinite, then in any partition of $V(G)$ into components of size at most k, $\mathcal{P}'$, at least $\epsilon' n = 4\delta_1 dn$ edges are cut by $\mathcal{P}'$. In particular, the local partition oracle of Lemma 4.3 induces a partition to components of size at most k. Hence, random query will be marked with probability of at least $4\delta_1 dn \cdot \frac{2}{dn} \geq 8\delta_1$ to be cut. Again, by Chernoff, the probability that at most a $4\delta_1$-fraction of the queries are marked is at most 1/20. □

We now formally complete the proof of the Theorem. If $G \in \Pi \cap \mathcal{C}$, then by Claim 7.1 it passes Phase 1 with probability 0.875. Since Phase 2 is just the tester for the proof of Theorem 3.3, it passes this test with probability at least 4/5. Hence G is accepted by the whole test with probability at least $0.675 \geq 2/3$.

Now consider a graph G that is ϵ-far from Π. If the graph is not (ϵ', k)-hyperfinite, it will be rejected with probability at least 9/10 in Phase 2. Thus, we can assume that it is (ϵ', k)-hyperfinite. Now Theorem 3.3 implies that G is rejected. □

8. Testing Isomorphism of Hyperfinite Graphs. We now present a short proof for testing isomorphism in a hyperfinite family. We first restate the result.

Theorem 8.1 (Isomorphism of hyperfinite graphs is testable; restated). *Let $\mathcal{C}$ be a ρ-hyperfinite family of d-bounded degree graphs. Then there is a function $s = s(\epsilon, \rho, d)$ such that for any $0 < \epsilon < 1$ there is a tester $A_{\epsilon,d}$ for graph isomorphism for two graphs in $\mathcal{C}$ whose query complexity is s. Namely, the algorithm $A_{\epsilon,d}$ has access to two graphs $G_1, G_2 \in \mathcal{C}$ with $|V_1| = |V_2|$. It makes at most s queries to G_1*

and G_2 and accepts G_1, G_2 with probability at least $2/3$, if G_1 is isomorphic to G_2, and rejects with probability at least $2/3$ if G_1 is ϵ-far from any isomorphic copy of G_2. If the function ρ is computable, then there is also a uniform tester for graph isomorphism.

Proof. Let $\epsilon > 0$, $\mathcal{C}$ be a ρ-hyperfinite family of graphs, and $G_1, G_2 \in \mathcal{C}$ be two d-bounded degree graphs for which we want to ϵ-test for isomorphism. The idea is quite simple, we just use the oracle of Lemma 5.2 to compute an appropriate estimate of the k-discs frequency vectors $\tilde{f}_1, \tilde{f}_2$ for partitions of G_1, G_2 respectively. We then accept if the graph described by $\tilde{f}_1$ is close enough to the graph described by $\tilde{f}_2$ and otherwise reject.

Formally, let $\delta = \lambda = \epsilon/8$, $\epsilon' = \epsilon_{5.2}(\delta)$ and let $k = \rho(\epsilon')$. We run the randomized algorithm guaranteed by Lemma 5.2, independently on G_1 and G_2 with parameters δ, λ, k as above. Due to the choice of parameters, and the assumption that $G_1, G_2, \in \mathcal{C}$ which is ρ-hyperfinite, these two runs produce two vectors $\tilde{f}_1, \tilde{f}_2$ for G_1, G_2 respectively, that with probability $3/5$ the following events occur simultaneously: (a) The partition $\mathcal{P}_1$, and the partition $\mathcal{P}_2$ that are induced by the oracle runs on G_1, G_2 respectively are $(\delta d, k)$-partitions. (b) $\|\tilde{f}_i - f_i^*\|_1 \leq \lambda,\ i = 1, 2$, where f_i^* is the k-discs frequency vector of $G_i[\mathcal{P}_i],\ i = 1, 2$. We accept, if there exist graphs H_1, H_2 with frequency vectors $\hat{f}_1, \hat{f}_2$ such that H_1 is 2δ-close to H_2 and $\|\tilde{f}_i - \hat{f}_i\|_1 \leq \lambda$ for $i = 1, 2$.

Assume first that G_1 is isomorphic to G_2. We will define $H_i = G_i[\mathcal{P}_i]$, $i = 1, 2$, and show that our choice of H_i leads to acceptance. Assume that events (a) and (b) above occur (which happens with probability at least $3/5$). Event (b) implies that $\|\tilde{f}_i - \hat{f}_i\|_1 \leq \lambda$ and event (a) implies that H_i is δ-close to $G_i,\ i = 1, 2$. It follows, that H_1 is 2δ-close to H_2 since G_1 and G_2 are isomorphic. Hence, the tester accepts with probability at least $3/5$.

Assume now that events (a) and (b) occur and that the tester accepts. In this case, we know by the acceptance condition that there are frequency vectors $\hat{f}_i,\ i = 1, 2$, describing graphs H_i such that H_1 is (2δ)-close to H_2. Furthermore, we know by the acceptance condition that $\|\hat{f}_i - \tilde{f}_i\|_1 \leq \lambda$ for $i = 1, 2$ and by condition (b) that $\|\tilde{f}_i - f_i^*\|_1 \leq \lambda$. Thus, $\|\hat{f}_i - f_i^*\|_1 \leq 2\lambda$ and Claim 2.2 implies that H_i is (2λ)-close to $G_i[\mathcal{P}_i]$. Hence, $G_1[\mathcal{P}_1]$ is $(2\delta + 4\lambda)$-close to $G_2[\mathcal{P}_2]$.

By the fact that $\mathcal{P}_1, \mathcal{P}_2$ are $(\delta d, k)$-partitions, it now follows that G_1 is $(4\delta + 4\lambda)$-close to G_2. By our choice of δ, λ this implies that G_1 is ϵ-close to G_2. Hence the test is correct with probability $3/5$ which can be amplified by repetition in a standard way. □

9. The local view of hyperfinite graphs approximates its global structure. Finally, we prove the pure combinatorial result of Theorem 3.1, which we will first restate below.

THEOREM 9.1 (local versus global graph structure; restated). *Let G_1, G_2 be d-bounded degree, ρ-hyperfinite graphs on n vertices. Then for every ϵ, $0 < \epsilon \leq 1$, there exists $\eta = \eta(\epsilon, \rho, d)$, $D = D(\epsilon, \rho, d)$, such that, if $\|freq_{G_1}(D) - freq_{G_2}(D)\|_1 \leq \eta$ then G_1 is ϵ-close to being isomorphic to G_2.*

Proof. We prove the theorem for $n \geq N$, for some constant number N and obtain a value D' for which the theorem holds. If we then take $D = \max\{D', N\}$ and $\eta < 1/N$ the theorem follows for every n, which can be seen as follows. If $n < N$ the graphs must be identical since each disc completely covers the connected component of its vertex and by our choice of η the frequency vectors are identical. If $n \geq N$ and the two graphs satisfy $\|freq_{G_1}(D) - freq_{G_2}(D)\|_1 \leq \eta$ they also satisfy $\|freq_{G_1}(D') - freq_{G_2}(D')\|_1 \leq \eta$

since $\|freq_{G_1}(D)-freq_{G_2}(D)\|_1 \geq \|freq_{G_1}(D')-freq_{G_2}(D')\|_1$. Thus, the result follows already from the case $n \geq N$ and disc radius D'.

Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ be d-bounded degree ρ-hyperfinite graphs on n vertices, let $\epsilon > 0$ and $k = \rho(\epsilon/4)$. Furthermore, assume that G_1 and G_2 satisfy $\|freq_{G_1}(D) - freq_{G_2}(D)\|_1 \leq \eta$ for $\eta = \eta(\epsilon, \rho, d)$ and $D = D(\epsilon, \rho, d)$ to be determined below. In order to show that G_1 is ϵ-close to G_2 we will show that they both have partitions $\mathcal{P}_1, \mathcal{P}_2$ respectively, that are $(\epsilon d/4, k)$-partitions, and with frequency vectors of their connected components (which are also k-discs) that are $\epsilon/2$-close. This will imply that G_1 is ϵ-close to G_2 by the triangle inequality: G_i is $\frac{\epsilon}{4}$-close to $G_i[\mathcal{P}_i]$, $i = 1, 2$ by definition, and $G_1[\mathcal{P}_1]$ is $\frac{\epsilon}{2}$-close to $G_2[\mathcal{P}_2]$ by Claim 2.2.

To show that there are corresponding partitions with close frequency vectors we will use the algorithm associated with Lemma 5.2, applied on both graphs, with parameters $\epsilon/4, \lambda = \epsilon/4$ and k as above. The choice of parameters here also determines our choice for D, namely, $D = D_{5.2}(\lambda, k, d)$. With probability at least 4/5 the algorithm will return an estimate of the frequency vector of the connected components of an $(\epsilon d/4, k)$-partition with ℓ_1-error at most $\epsilon/4$. We recall that this algorithm samples a set S of vertices uniformly at random and then explores the D-discs around them. It uses its internal randomness to (a) determine the set S and (b) to determine the behavior of the partitioning oracles. Note that, if for two distinct graphs G_1 and G_2 the algorithm queries the same set of D-discs and if these D-discs do not intersect in any of the graphs, then the output distribution (which still depends on the choice of randomness for the partitioning oracle) is identical.

Since $\|freq_{G_1}(D) - freq_{G_2}(D))\|_1$ is bounded by η, but not necessarily 0, we do not expect that for the sample S, the algorithm will see exactly the same set of discs in both graphs, but mostly so. To show that this is enough we need the following properties of the sampled sets.

Let $s = s_{5.2}(\lambda, k, d)$ and $\eta = \frac{1}{10s}$. We first fix a mapping $\Psi : V_1 \mapsto V_2$ that identifies the vertices of V_1 with the vertices of V_2 such that for all $v \in V_1$, but an η-fraction of the vertices, $B_{G_1}(v, D)$ is isomorphic to $B_{G_2}(\Psi(v), D)$. This is asserted by Claim 2.1, using the fact that $\|freq_{G_1}(D) - freq_{G_2}(D)\|_1 \leq \eta$. Let $F \subseteq V_1$ be the set of these vertices v for which $B_{G_1}(v, D)$ is not isomorphic to $B_{G_2}(\Psi(v), D)$. Then $|F| \leq \eta \cdot n$. We want to construct a sample set $S \subseteq V_1$, such that the following property is satisfied:

Definition 9.2. *A sample set $S \subseteq V_1$ is called* good*, if for some choice of random bits for the partitioning oracle,*

1. *the algorithm on input G_1 and sample set S approximates the frequency vector of an $(\epsilon d/4, k)$-partition of G_1 with ℓ_1-error at most λ,*
2. *the algorithm on input G_2 and sample set $\Psi(S) := \{\Psi(s) \mid s \in S\}$ approximates the frequency vector of an $(\epsilon d/4, k)$-partition of G_2 with ℓ_1-error at most λ, and*
3. *the output vector of both instances is identical.*

By the discussion above, if S is good, then indeed $G_1[\mathcal{P}_1]$ is $\epsilon/2$-close to $G_2[\mathcal{P}_2]$ as needed, where $\mathcal{P}_i$, $i = 1, 2$ are $(\epsilon/4, k)$-partition for G_i. Hence, we only need to show the existence of good S.

Definition 9.3. *We say that a sample set $S \subseteq V_1$ is D-*indistinguishable *in G_1 and G_2, if the following two conditions are satisfied:*

- *all D-discs in G_1 rooted at vertices in S do not intersect,*
- *all D-discs in G_2 rooted at vertices $\Psi(v)$, $v \in S$ do not intersect, and*
- *$S \cap F = \emptyset$.*

To show that there exists a good sample set S, we will focus on D-indistinguishable sample sets. If a sample set S is D-indistinguishable, we can easily extend any permutation of V_1 to a permutation of V_2 in such a way that, on input S and $\Psi(S)$ and with randomness given by the two corresponding permutations, the output vectors returned by the two instances of our algorithm will be identical. Furthermore, if we take the permutation of V_1 as random, so will be the permutation of V_2 and so the behaviour of the algorithm on input V_2 will satisfy the same guarantees as that of a normal instance. If we can prove that for some set $S \subseteq V_1$ there exists a permutation of V_1 that satisfies Item 1 in Definition 9.2 and its corresponding permutation of V_2 satisfies Item 2, then we have proved the existance of a good sample set. We say that a sample set S is *sufficient* if this sufficient condition for being good is met for S.

In order to prove that a sufficient set S exists, we will first show that most sets S are D-indistinguishable for sufficiently large n. This follows, because the probability for the event $S \cap F \neq \emptyset$ is bounded by $2|S| \cdot \eta \leq 1/5$. Furthermore, as $|S|, k$ and D are independent of n, there exists $N = N(\epsilon, \rho, d)$ such that for all $n \geq N$ the probability that D-discs around the sampled vertices and their image under Ψ in G_2 intersect in G_1 or G_2 is at most $1/5$. Hence, with probability at least $3/5$ the set S is D-indistinguishable for $n \geq N$.

Now, recall that the algorithm in Lemma 5.2 queries a random sample set S, and S is successful for the algorithm with probability at least $9/10$. Moreover, when it is successful, it has Property 1 in Definition 9.2. Similarly, the same is true for $\Psi(S)$ to be successful for the run on G_2. Thus the probability that S is D-indistinguishable for G_1, G_2 and that it is sufficient is at least $1 - 3/5 - 2/10 > 0$. In particular this implies the existence of S that is sufficient. □

We end with the following discussion on the last proof. The proof presented above requires only Lemma 5.2 and hence it is purely combinatorial. As already mentioned the results presented are essentially equivalent. The isomorphism tester is immediately implied by Theorem 3.1 and vice versa, if the algorithm in the proof of Theorem 3.2 is also taken into account. The isomorphism tester can be derived from the two other testers by defining a graph property that contains only all isomorphic copies of graph H (the graph for which isomorphism is tested). Alternatively, the two other testers can also be derived from Theorem 3.1.

In the light of this discussion, let us remark that there is an even simpler proof for Theorem 3.1, but that is based on Theorem 3.2. We sketch this proof here: Let G_1, G_2 be two graphs that meet the assumptions of Theorem 3.1 (with the parameters say, set in its proof). Let us consider G_1, G_2 as the input for the ϵ-test for isomorphism that is asserted by Theorem 3.2. Since the test explores D-discs around a constant number of random vertices, the assumption on G_1, G_2 together with Claim 2.2 implies that the test must accept with probability $2/3 - 1/10$, as it sees isomorphic view with probability, say $9/10$ (as argued in the first part of the proof above). Hence, by the correctness of the test, it follows that G_1 must be ϵ-close to G_2.

10. Additional discussion. In view of the grand goal of the characterization of all graph properties that are testable in the bounded-degree graph model, and the discussion in the introduction, we note that all the testable properties that we are aware of, fall into one of the following categories: (a) the properties of hyperfinite classes, namely that are covered by Theorem 3.4[5]. (b) properties that are defined by

[5]Needless to say, most if not all of previously constructed testers are much more efficient than these implied by Theorem 3.4 here.

their local structure: e.g., being triangle-free. Such properties are testable for every bounded degree graph. (c). combinations of the above by simple boolean operators that 'preserve' the distance (e.g., the union of a property from (a) and a property of type (b)). (d) - other properties, such as, the property of having between $d/4$ to $d/2$ edges. Another such property is connectivity.

The last category contains testable properties for which testability is not directly due to any 'general' reason known so far. Finding a reasonable general explanation to why such properties are testable (besides the 'trivial' fact that their existence is determined by the frequency vector of the graph for some suitable disc-radius), would essentially amount to a characterization of testable properties in this model.

REFERENCES

[1] N. Alon, E. Fischer, I. Newman, A. Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)* pp. 251-260, 2006.

[2] N. Alon, J. Spencer. *The Probabilistic Method*, John Wiley, 1992.

[3] N. Alon, P. Seymour, R. Thomas. A separator theorem for graphs with an excluded minor and its applications. *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 293-299, 1990.

[4] I. Benjamini, O. Schramm, and A. Shapira. Every minor-closed property of sparse graphs is testable. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 393–402, 2008.

[5] C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, B. Szegedy, Katalin Vesztergombi. Graph limits and parameter testing. *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 261-270,2006.

[6] A. Czumaj, A. Shapira, and C. Sohler. Testing hereditary properties of nonexpanding bounded-degree graphs. *SIAM Journal on Computing*, 38(6): 2499–2510, 2009.

[7] G. Elek. L^2-spectral invariants and convergent sequences of finite graphs. *Journal of Functional Analysis*, vol. 254, no. 10, pp. 2667–2689, 2008.

[8] G. Elek. Parameter testing with bounded degree graphs of subexponential growth. *Random Structures & Algorithms*, 37(2): 248-270, 2010.

[9] G. Elek. Finite graphs and amenability. arXiv:1204.0449v2, 2012.

[10] O. Goldreich, S. Goldwasser, D. Ron. Property Testing and its Connection to Learning and Approximation. *Journal of the ACM*, 45(4): 653-750, 1998.

[11] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2): 302–343, 2002.

[12] A. Hassidim, J. A. Kelner, H. N. Nguyen, and K. Onak. Local graph partitions for approximation and testing. *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 22–31, 2009.

[13] R. Lipton and R. Tarjan. Applications of a Planar Separator Theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

[14] E. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1): 42-65, 1982.

[15] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, Vol. 298. no. 5594, pp. 824 - 827, 2002.

[16] G. Miller. Isomorphism of graphs which are pairwise k-separable. *Information and Control*, 56(1-2): 21–33, 1983.

[17] H. Nguyen and K. Onak. Constant-Time Approximation Algorithms via Local Improvements. *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 327–336, 2008.

[18] I. Newman and C. Sohler. Every Property of Hyperfinite Graphs is Testable. *Proceedings of the . 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 675-684, 2011.

[19] K. Onak. New Sublinear Methods in the Struggle Against Classical Problems. *PhD Thesis*, Massachusetts Institute of Technology, 2010.

[20] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal of Computing*, 25 (1996), 252–271 (first appeared as a technical report, Cornell University, 1993).

[21] S. Raskhodnikova, A. Smith. A Note on Adaptivity in Testing Properties in Bounded Degree

Graphs. *Electronic Colloquium on Computational Complexity*, TR06-089, 2006.