Every Property of Hyperfinite Graphs is Testable

[Extended Abstract]

Ilan Newman [*]
Department of Computer Science
University of Haifa, Haifa, Israel
ilan@cs.haifa.ac.il

Christian Sohler [†]
Department of Computer Science
Technische Universität Dortmund
christian.sohler@tu-dortmund.de

ABSTRACT

A property testing algorithm for a property Π in the bounded degree graph model[7] is an algorithm that, given access to the adjacency list representation of a graph $G = (V, E)$ with maximum degree at most d, accepts G with probability at least $2/3$ if G has property Π, and rejects G with probability at least $2/3$, if it differs on more than ϵdn edges from every d-degree bounded graph with property Π. A property is *testable*, if for every ϵ, d and n, there is a property testing algorithm $A_{\epsilon,n,d}$ that makes at most $q(\epsilon, d)$ queries to an input graph of n vertices, that is, a non-uniform algorithm that makes a number of queries that is independent of the graph size.

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 large enough number of vertices, n, 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.

We use this result to obtain new results and improve upon existing results in the area of property testing. In particular, 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.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

Keywords

Property Testing

General Terms

Algorithms, Theory

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 graph 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 k-separable graphs, which include all graphs whose maximum degree is bounded by a constant [11]). Assume 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 [7]. 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 ϵ-far from being isomorphic[2].

[*] This Research was supported by The Israel Science Foundation (grant number 862/10.)

[†] This research was supported by the German Research Foundation (DFG), grant So 514/3-2.

STOC'11, June 6–8, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0691-1/11/06 ...$10.00.

[1] the class of planar graphs here can be replaced with any hyper-finite class, which in turn, contains any class that is defined by a set of forbidden minors

[2] ϵ-far here means that one needs to modify at least ϵ-fraction of the edges of each graph so that the resulting graphs be-

A consequence of this main technical result, is that for *any* graph property $\mathcal{P}$, deciding whether a given planar graph has $\mathcal{P}$ or is ϵ-far from having $\mathcal{P}$, can be decided by $q = q(\epsilon, d)$-queries - a constant number of queries to the graph (independent of the graph size). Combining this with a result of Hassidim, Kelner, Nguyen, and Onak [8] (or Benjamini, Schramm and Shapira [2]) 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 hyper-finite graph property. Previously, this was only known for hereditary hyper-finite graph properties, which contain, for example, all minor-closed graph properties [8] (see also [2] for such a result for monotone graph properties).

Another immediate corollary of our result 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 hyper-finite graphs. [3] Previously, this was known only for specific functions like maximum matching, minimum vertex cover, minimum dominating set and maximum independent set [12, 8, 5].

In the next couple of paragraphs we briefly describe the recent relevant developments in property testing of bounded degree graphs. As already noted in [2], our understanding of property testing in the dense graph model is much better than that for the bounded-degree one. This is since Szemerédi 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 theory.

On the other hand, it is clear what a constant query tester can do: it can sample some constant number of vertices from the graph, and explore a constant neighborhood around these vertices. Then, the tester needs to make its decision based on the local view it finds (and possibly its internal randomness). Looking at the distribution such a tester induces by its local coins on the sampled subgraphs, this means that its decision is made by looking at the frequency of appearance of constant size subgraphs in the graph. To understand what properties can be tested in this model, one needs to understand what the frequency of appearance of small subgraphs can tell us about the the whole graph. The first step in this direction was done by Czumaj, Shapira and Sohler in [4], 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 [2], 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 having 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 the membership in any monotone hyperfinite property of graphs is testable in constant time. In a follow-up work, Hassidim, Kelner, Nguyen, and Onak [8] give an explicit algorithm to locally compute such a partition. They prove that every hyperfinite hereditary graph property is testable and also give a simpler proof for testing minor-closed properties (a subclass of monotone graph properties) and improve the running time of the test. 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 [8] we show that this local view is sufficient for any property of planar graphs.

Our techniques heavily rely on the recent development in the area of testing in the bounded degree model. In particular, we heavily use the local-partition oracles used implicitly in [2], and explicitly in the later result by Hassidim, Kelner, Nguyen, and Onak [8], which in turn is based on a technique developed in [12]. Our main argument uses the probabilistic method to show that two graphs that have a similar distribution of local neighborhoods are close to be isomorphic. In order to do so, we consider two instances of a simple sampling algorithm to estimate the frequency of connected components in a partition provided by a local-partitioning oracle [8]. We couple the randomness used by the two instances to show that with positive probability they return the same (good) estimate of these frequencies. Since the local partitions can be obtained by removing, say $\epsilon dn/3$ edges from each of the graphs, and since two graphs that consists of small connected components with similar frequencies can be transformed into each other by changing, say, $\epsilon dn/3$ edges, it follows that the resulting graphs are ϵ-close.

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 [10]. Motifs are subgraphs that occur significantly more frequent in a given class of graphs than in a random graph. They are used to classify certain classes of networks. To put it differently, 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.

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 [8] along with an estimator for the frequency vector of local neighborhoods. Section 6 contains the proof of the main technical result and finally Section 7 contains the proofs of the corollaries of the main theorem, along with an additional discussion.

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

come isomorphic. We will refer to such algorithms as *property testing algorithms* or *testers*, for short. See exact definitions in Section 2

[3] The algorithms for such approximations are non-uniform in nature, hence have no uniformly bounded time complexity. However, this is to be expected for a statement of this generality.

vertex set V and edge set E. We write $V(G)$ to denote the vertex set of graph G, and will always assume that $V(G) = \{1, \ldots, n\}$ for the graph G at hand. We will shortly say that a graph is d-degree bounded, 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). A d-degree bounded graph will be represented by its adjacency lists. This can be thought of 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* [7].

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be *isomorphic*, if there is a mapping $\Phi : V_1 \to V_2$ such that $(u, v) \in E_1$, if and only if $(\Phi(u), \Phi(v)) \in E_2$. A graph property is a (possibly infinite) collection of graphs, which is closed under isomorphism.

Definition 1. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be d-degree bounded graphs. The distance $dist(G_1, G_2)$ is the amount 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 (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.*

Definition 2 (ϵ-far). *Let Π be any (non-empty) graph property on d-degree bounded graphs. A d-degree bounded 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.*

The notion of property testing was introduced by Rubinfeld and Sudan [14] and then formally defined by Goldreich, Goldwasser and Ron [6]. A property testing algorithm for property Π, or, for short, property tester, in the bounded degree graph model, is an algorithm that, given query access to a graph G in the bounded degree graph model, 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, the algorithm can answer arbitrarily. A query[4] in this model 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 query complexity of a tester is the number of queries for the worst case input (and worst case random choices).

Definition 3 (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.*

2.1 Partitions and frequencies of subgraphs

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

A connected graph $G = (V, E)$ with a specially marked vertex v, is called *rooted graph* and we sometimes say that G is rooted at v. A rooted graph $G = (V, E)$ 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 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 with root v. $B_G(v, k)$ 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 possible non-isomorphic k-discs is at most $N(k, d)$.

The main results here roughly state that knowing the number of each type of disc in a planar graph (or even an approximation of it) 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 4. *For a d-bounded degree graph $G = (V, E)$ and integer k, let $\mathrm{dist_G(k)}$ be the distribution vector of all k-discs of G. Namely, $\mathrm{dist_G(k)}$ is a vector of dimension $N(k, d)$, indexed by all possible rooted graphs of radius at most k and degree at most d. Each entry of $\mathrm{dist_G(k)}$ corresponds to some fixed rooted graph H, and counts the number of k-discs of G that are isomorphic to H. Note that G has $n = |V|$ different discs, thus the sum of entries in $\mathrm{dist_G(k)}$ is n. Let $\mathrm{freq_G(k)}$ be the normalized distribution, namely $\mathrm{freq_G(k) = dist_G(k)/n}$.*

For vector $v = (v_1, \ldots, v_r)$ we will use $\|v\|_1 = \sum_{i=1}^{r} |v_i|$ to denote its l_1-norm. Note that for two vectors $u = (u_1, \ldots, u_r)$, $v = (v_1, \ldots, v_r)$ representing distributions on r elements, i.e. $\|u\|_1 = \|v\|_1 = 1$, $\|v - u\|_1$, is exactly two times the variation distance between the two distributions. We say that two unit-length vectors u, v are λ-close, if $\|u - v\|_1 \leq \lambda$.

Finally to state our results we need the following definition of *hyperfinite* graphs, which was introduced in [5]. We note that all planar graphs, as well as any graph family that is defined by a set of forbidden minors is hyperfinite.

Definition 5 (Hyperfinite). *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 $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ we call a collection of graphs ρ-hyperfinite, if, for every $\epsilon > 0$, every graph in the collection is $(\epsilon, \rho(\epsilon))$-hyperfinite.*

3. RESULTS

Our main technical tool is the following theorem.

[4] A query in this model is usually for a vertex $v \in V$ and a number i, to which 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, for $d = O(1)$, we assume that a query just specifies a vertex $v \in V$, to which the return is the set of all (at most d) neighbors of v.

THEOREM 3.1. *There exists functions $\epsilon_{3.1} = \epsilon_{3.1}(\epsilon)$, $N_{3.1} = N_{3.1}(\epsilon, k, d), D_{3.1} = D_{3.1}(\epsilon, k, d), \lambda_{3.1} = \lambda_{3.1}(\epsilon, k, d)$ such that for every $0 < \epsilon < 1$ and $k \geq 1$ the following holds: For every two $(\epsilon_{3.1}, k)$-hyperfinite graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ on $n \geq N_{3.1}$ vertices and with maximum degree at most d, if $|\mathrm{freq}_{G_1}(D_{3.1}) - \mathrm{freq}_{G_2}(D_{3.1})| \leq \lambda_{3.1}$ then G_1 is ϵ-close to G_2.*

A corollary of Theorem 3.1 is the following theorem.

THEOREM 3.2. *Let $\mathcal{C}$ be a ρ-hyperfinite family of graphs with maximum degree at most d. Then there is a function $s = s_{3.2}(\epsilon, \rho, d)$ 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.*

Our main technical theorem implies that every planar (or, more generally, every hyperfinite) d-degree bounded graph is determined up to ϵdn edges by the distribution of D-discs (of certain size). For a fixed n and ϵ, we can represent a property Π by the set of distribution vectors of the graphs having Π. Then we can discretize the space and mark every vector which has a nearby vector in Π. This way, we get a finite representation of the property. An algorithm can now simply estimate the frequency vector of the D-discs and move to the nearest one in the discretized space. It accepts if and only if this nearest frequency vector belongs to the property. This leads to the following theorem.

THEOREM 3.3. *Let $\mathcal{C}$ be a ρ-hyperfinite family of graphs, and $\mathcal{P}$ any graph property. The property $\mathcal{P}$ is testable for any graph taken from $\mathcal{C}$.*

Finally, using the testability of a necessary condition for hyperfiniteness in a similar way as in the tester for hereditary properties from [8], this implies the following theorem.

THEOREM 3.4. *Let $\mathcal{C}$ be a ρ-hyperfinite family of graphs that is closed under isomorphism, and $\mathcal{P}$ any graph property, then for any graph in the bounded graph model, the property of being in $\mathcal{C}$ and having $\mathcal{P}$ is testable.*

Another 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 $\mathcal{P}(\ell, \epsilon)$ be the set of all graphs G for which $\ell - \epsilon d|V| \leq f(G) \leq \ell$. Since $\mathcal{P}(\ell, \epsilon)$ is a graph property, Theorem 3.3 asserts that $\mathcal{P}(\ell, \epsilon)$ 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 with in an additive error of $\epsilon d|V|$ by testing $\mathcal{P}(\ell, \epsilon)$ for various $\ell's$. Thus we get,

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

4. ALGORITHMS FOR LOCAL GRAPH PARTITIONING

Every graph taken from a hyperfinite family of graphs admits a partitioning into small connected components. 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 [8]. Also, an earlier work of Benjamini, Schramm and Shapira[2] 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 [12] that can be used to derive constant time algorithms from certain greedy approximation algorithms. In [8] the authors present a greedy algorithm to compute the desired partition, which then can be made local using the technique from [12]. 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 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. More precisely, we call a set $S \subseteq V$ a (k, η)-*isolated neighborhood* of $v \in V$, if $v \in S$, the subgraph induced by S is connected, $|S| \leq k$ and $e(S) \leq \eta|S|$. With this definition, we can state the algorithm from [8] that computes a partition of G.

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

Note that the random permutation of the vertices deterministically defines the partition. We thus denote this partition as $\mathcal{P}(\pi)$. The partition, of course, depends also on η 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 such a partition are of size at most k. If in addition at most ϵn edges are cut, we say that the partition is an (ϵ, k)-partition.

It is proved in [8] that applying the randomized algorithm above for G, taken from a ρ-hyperfinite family of graphs, with proper choice of η and k as functions of ϵ, results, with probability at least $9/10$, in 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 [8]. Recall that in this paper a query of a graph vertex $v \in V(G)$ returns all of the neighbors of v. This is geared towards property testing in the bounded degree graph model, in which such a query can be implemented in $O(1)$ queries and time.

Definition 6. *[8] We say that $\mathcal{O}$ is a (randomized) (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 'name' of the partition class that contains v. The partition has the following properties:*

- *$\mathcal{P}$ is a function of the graph and random bits of 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 $9/10$.*

It is shown in [8] how to implement the global partition above locally, so to result in a local partition 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[5]. The advantage of computing such a 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.

The local computation of a random map π as above can be used to locally construct the partition class of a queried vertex v in the following way: We explore the $2k$-disc around v and whenever we encounter a new vertex whose priority has not been fixed, we choose its priority uniformly at random from $[0, 1]$. If v has lowest priority among all discovered vertices in the neighborhood, we know that it would be cut out by the algorithm GlobalPartitioning, first in his k-disc. 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. Otherwise, if the neighborhood around v does contain a vertex u with $\pi(u) < \pi(v)$, then we recurse by moving to u with the smallest priority.

We 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 [8] that the expected number of queries (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 [8] (see also [13]).

Lemma 4.1. *[8] Let G be an $(\epsilon^3/54000, k)$-hyperfinite graph with degree bounded by $d \geq 2$. 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 the D-discs with radius bounded by $D = D_{3.1}(\epsilon, d)$ in a graph $G = (V, E)$. Let $\{H_1, \ldots, H_{N(D,d)}\}$ denote the set of all d-bounded degree rooted graphs with radius at most D (in particular this includes all D-discs that are isomorphic to the D-discs of a particular d-bounded degree graph). We assume that H_i corresponds to the i-th entry in the frequency vector $\mathrm{freq_G(D)}$, i.e. the entry gives the relative frequency of H_i in G. The vector $\mathrm{freq_G(D)}$ can be estimated to within additive error $\pm\lambda$ by sampling a constant number of vertices and exploring the D-discs around them.

Lemma 5.1. *Let G be d-degree bounded, and $s \geq \frac{N(D,d)^2}{\lambda^2} \cdot \ln(20N(D, d))$. The algorithm* EstimateFrequencies *below samples s vertices uniformly at random, explores their D-discs and with probability at least $9/10$ returns a vector $\widetilde{\mathrm{freq}}_G(D)$ with $\|\widetilde{\mathrm{freq}}_G(D) - \mathrm{freq_G(D)}\|_1 \leq \lambda$.*

```
EstimateFrequencies($G = (V, E), D, s$)
    $\widetilde{\mathrm{freq}}_G(D) = 0$
        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 $H_i$ be the
            resulting induced rooted subgraph
            set $\widetilde{\mathrm{freq}}_G(D)[i] = \widetilde{\mathrm{freq}}_G(D)[i] + \frac{1}{s}$
    return $\widetilde{\mathrm{freq}}_G(D)$
```

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 $\mathrm{freq_G(D)[i]}$. This implies that $\mathbf{E}[X_{i,j}] = \mathrm{freq_G(D)[i]}$ and $\mathbf{E}[\sum_{j=1}^{s} X_{i,j}] = s \cdot \mathrm{freq_G(D)[i]}$. Furthermore, we have $\mathbf{E}[\widetilde{\mathrm{freq}}_G(D)[i]] = \frac{1}{s} \cdot \mathbf{E}[\sum_{j=1}^{s} X_{i,j}] = \mathrm{freq_G(D)[i]}$.

Using Chernoff bounds we get for every i,

$$\begin{aligned} & \mathbf{Pr}[|\widetilde{\mathrm{freq}}_G(D)[i] - \mathrm{freq_G(D)[i]}| \leq \mathrm{\lambda/N(D,d)}] \\ = \; & \mathbf{Pr}[|\sum_{j=1}^{s} X_{i,j} - \mathbf{E}[\sum_{j=1}^{s} X_{i,j}]| \leq \frac{\lambda}{N(D,d)} \cdot s] \\ \leq \; & 2e^{-2\frac{\lambda^2}{N(D,d)^2}s} \\ \leq \; & \frac{1}{10N(D,d)} \end{aligned}$$

By the union bound we get that with probability at least $9/10$ for every i, $|\widetilde{\mathrm{freq}}_G(D)[i] - \mathrm{freq_G(D)[i]}| \leq \mathrm{\lambda/N(D,d)}$. This implies that $\|\widetilde{\mathrm{freq}}_G(D) - \mathrm{freq_G(D)}\|_1 \leq \lambda$. ■

[5] 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.

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 the connected components of $G[\mathcal{P}]$, 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 provides access to $G[\mathcal{P}]$, as asserted by Lemma 4.1. Namely, each query in the frequency estimate is done by calling the partition oracle. Note that these queries are non-adaptive, and hence the guarantees of Lemma 4.1 hold. Since every connected component in $G[\mathcal{P}]$ has diameter at most k, we know that every k-disc corresponds to a connected component of $G[\mathcal{P}]$.

This is formally stated in the following Lemma.

Lemma 5.2. *For every choice of constants ϵ, λ with $0 < \epsilon, \lambda < 1$ and every $k \geq 1$, there are values $\epsilon_{5.2} = \epsilon_{5.2}(\epsilon)$, $D_{5.2} = D_{5.2}(\epsilon, k, d)$, $s_{5.2} = s_{5.2}(\epsilon, \lambda, k, d)$ and a randomized algorithm* Sampler*, that on a random permutation π (given as a priority vector), accesses the graph G 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 $(\epsilon_{5.2}, k)$-hyperfinite and with degree bound $d \geq 2$, 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 an $(\epsilon d, k)$-partition.*

2. *The output vector $\widetilde{f}$ approximates the frequency vector f of the k-discs in $G[\mathcal{P}(\pi)]$. Namely, we have $|\widetilde{f} - f|_1 \leq \lambda$.*

We note that the Lemma asserts the existence of a randomized algorithm (the "sampler") whose output is a frequency vector $\widetilde{f} = \widetilde{f}(\pi, S)$, that depends on the internal randomness: the random permutation π and the sampled s-sequence of vertices S, where an s-sequence of vertices from V is just a member of V^s, and $s = s_{5.2}$. This frequency vector estimates the frequency vector of the partition $\mathcal{P}(\pi)$ that is defined only by π, and that is an $(\epsilon d, k)$-partition with high probability (over choices of π). The probability in the lemma is with respect to both choices of π and S.

We also note that we never need in our testers to apply the algorithm Sampler of Lemma 5.2. However, the existence of such an estimator is crucial for the proof of our main Theorem.

6. PROOF OF THEOREM 3.1

Let $0 < \epsilon < 1$, $k \geq 1$ and let us define the functions from Theorem 3.1 as follows: $\epsilon_{3.1}(\epsilon) := \epsilon_{5.2}(\frac{\epsilon}{4})$, $D_{3.1}(\epsilon, k, d) := D_{5.2}(\frac{\epsilon}{4}, k, d)$, and $s_{3.1}(\epsilon, k, d) := s_{5.2}(\frac{\epsilon}{4}, \frac{\epsilon}{4}, k, d)$. Further define $\lambda_{3.1} := \frac{1}{10 s_{3.1}}$. Let $\epsilon_{3.1} = \epsilon_{3.1}(\epsilon)$, $D_{3.1} = D_{3.1}(\epsilon, k, d)$, and $s_{3.1} = s_{3.1}(\epsilon, k, d)$. Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ be two $(\epsilon_{3.1}, k)$-hyperfinite graphs with $V_1 = V_2 = \{1, \ldots, n\}$ and for which $\mathrm{freq_{G_1}(D_{3.1})}$ and $\mathrm{freq_{G_2}(D_{3.1})}$ are $\lambda_{3.1}$-close, as in the premise of Theorem 3.1. We start with the following observations, which are true for any values of k, D, λ and in particular for the ones chosen here.

Claim 6.1. *Let $D \geq 0$ be an integer and $0 < \lambda < 1$. If $\mathrm{freq_{G_1}(D)}$ and $\mathrm{freq_{G_2}(D)}$ 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, $B_{G_1}(v, D)$ is isomorphic to $B_{G_2}(\Psi(v), D)$.*

Proof. By greedily mapping v to u if their D-discs are isomorphic, independent of previous choices. ■

Claim 6.2. *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 $\mathrm{freq_{G_1^*}(k)}$ and $\mathrm{freq_{G_2^*}(k)}$ are λ-close. Then G_1^* and G_2^* are λ-close.*

Proof. Map greedily 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 arbitrary. By definition, only λ-fraction of the vertices will be mapped to images on which the 1-neighborhoods do not agree. By the bounded degree assumption the claim follows. ■

The top level view of the proof is as follows. Since G_1, G_2 are both $(\epsilon_{3.1}, k)$-hyperfinite (and we may assume $\epsilon_{3.1} \leq \frac{\epsilon}{4}$), we know that they have $(\frac{\epsilon d}{4}, k)$ partitions, $\mathcal{P}_1, \mathcal{P}_2$, respectively. Thus, (by definition of $(\frac{\epsilon d}{4}, k)$ partition) $G_i^* = G_i[\mathcal{P}_i]$ is $\frac{\epsilon}{4}$-close to G_i, $i = 1, 2$. Moreover, since $\epsilon_{3.1} = \epsilon_{5.2}(\frac{\epsilon}{4})$, Lemma 5.2 guarantees that the global partition algorithm yields such partitions with high probability when applied with a random permutation (independently for G_1, G_2). We will show (this will be the main technical part, and only here we need to resort to the assumption that $\mathrm{freq_{G_1}(D_{3.1})}$ and $\mathrm{freq_{G_2}(D_{3.1})}$ are $\lambda_{3.1}$-close) that for certain such partitions $\mathcal{P}_1, \mathcal{P}_2$, for G_1, G_2 respectively, the frequency vectors $\mathrm{freq_{G_1^*}(k)}$ and $\mathrm{freq_{G_2^*}(k)}$ are $\frac{\epsilon}{2}$-close, where $G_1^* = G_1[\mathcal{P}_1]$ and $G_2^* = G_2[\mathcal{P}_2]$. Hence, by Claim 6.2 it follows that G_1^* is $\frac{\epsilon}{2}$-close to G_2^*. Composing this with the fact that G_i is $\frac{\epsilon}{4}$-close to G_i^*, $i = 1, 2$ implies that G_1 is ϵ-close to G_2.

To exhibit the existence of the corresponding partitions $\mathcal{P}_1, \mathcal{P}_2$, we first construct $\mathcal{P}_1$ by using the local partition oracle, for parameters $(\frac{\epsilon d}{4}, k)$, whose existence is asserted by Lemma 4.1 and that is accessed by algorithm Sampler from Lemma 5.2. The parameters $k_{3.1}, D_{3.1}$ and $s_{3.1}$ are chosen such that the sampler in Lemma 5.2 runs with parameters $k_{5.2} = k_{3.1}, D_{5.2} = D_{3.1}$ and $s_{5.2} = s_{3.1}$, and will output a good estimate $\tilde{f}_1$ of the real frequency vector f_1, of that partition. Namely, the sampler in Lemma 5.2 makes a non adaptive sample of $s_{3.1}$ random vertices, chooses a random permutation π on V_1, and constructs the output vector $\tilde{f}_1$ based only on the the $D_{3.1}$-discs around each of the sampled vertices and their relative order according to π. The output vector is so that $|f_1 - \tilde{f}_1|_1 \leq \frac{\epsilon}{4}$.

Thus the partition $\mathcal{P}_1(\pi)$ is a random partition that depends on the random permutation π, and the estimate $\tilde{f}_1$ depends on π and the $s_{3.1}$ sampled points. We will now hope that a run of the sampler on G_2 will produce an estimate $\tilde{f}_2$ with the analogous guarantees. Moreover, we hope to arrive at the situation where $\tilde{f}_1 = \tilde{f}_2$. This seems unlikely, since the estimate depends crucially on the local random choices, as explained above, and thus there is no reason to expect that f_1 will be any close to f_2 even if the graphs were isomorphic.

Now, to construct $\mathcal{P}_2$, so that the sampler will give a close output to its output for $\mathcal{P}_1$ on G_1, recall that by assumption, $\mathrm{freq}_{\mathrm{G}_1}(\mathrm{D}_{3.1})$ and $\mathrm{freq}_{\mathrm{G}_2}(\mathrm{D}_{3.1})$ are $\lambda_{3.1}$-close. Thus by Claim 6.1 there is a bijection that maps most vertices of G_1 to vertices of G_2 with corresponding isomorphic discs. Fix such a bijection $\Psi : V_1 \mapsto V_2$ as asserted in Claim 6.1. The existence of Ψ means that for a random vertex sampled in G_1, there is with high probability, a corresponding vertex in G_2 so that the corresponding local discs are isomorphic. If we could coordinate the random choices for the sampler run on G_2 with those for G_1 via the mapping Ψ, we would essentially be done, as the local views will be the same, and hence the output must be the same.

This suggests the following strategy. Recall that the sampler in Lemma 5.2 uses a permutation π on $V(G)$ that is distributed uniformly at random and a $s_{3.1}$-sequence of vertices $S \in V(G)^{s_{3.1}}$ which is distributed uniformly at random (on $V(G)^{s_{3.1}}$) and independent of π. Its output $\tilde{f}$ depends on π and S deterministically. Thus the run of the sampler on some arbitrary graph G can be viewed as using a sample from the product space $\mathcal{Q} \times \Pi$, where $\mathcal{Q}$ is the set of all $s_{3.1}$-sequences and Π is the set of all permutations of $\{1, \ldots, n\}$.

Let $\mathcal{Q}_i \times \Pi_i$, $i = 1, 2$ be the corresponding spaces for G_1, G_2. We will define a bijection $\Phi : (\mathcal{Q}_1 \times \Pi_1) \mapsto (\mathcal{Q}_2 \times \Pi_2)$. Observe that every bijection has the property that the uniform distribution on $\mathcal{Q}_1 \times \Pi_1$ induces, under Φ, the uniform distribution on $\mathcal{Q}_2 \times \Pi_2$. Thus given the random input $(S, \pi) \in \mathcal{Q}_1 \times \Pi_1$ for the run on G_1, we can use $\Phi(S, \pi)$ as the random input for the run on G_2, and from the point of view of the stand-alone algorithm run on G_2 this is a legitimate run. Hence, the sampler will produce a legitimate estimate $\tilde{f}_2$ of a corresponding partition on G_2. We will make sure (with high probability), using this coupling between the runs, that the local view defined by (S, π) on G_1 is isomorphic to the view defined by $\Phi(S, \pi)$ on G_2. Hence, both runs of the algorithm have the same local view and so they will produce the same output.

We now formally define the bijection $\Phi : (\mathcal{Q}_1 \times \Pi_1) \mapsto (\mathcal{Q}_2 \times \Pi_2)$ and present the whole proof. As a starting point, we fix a bijection $\Psi : V_1 \mapsto V_2$ as asserted in Claim 6.1 for $\lambda = \lambda_{3.1}$ and $D = D_{3.1}$. This bijection has the property that, for most vertices v, the $D_{3.1}$-disc $B_{G_1}(v, D_{3.1})$ rooted at v in G_1 is isomorphic to the $D_{3.1}$-disc $B_{G_2}(\Psi(v), D_{3.1})$ rooted at $\Psi(v)$ in G_2. This implies that if we pick an $s_{3.1}$-sequence of vertices $(v_1, \ldots, v_{s_{3.1}})$ from V_1, then the local view of the corresponding $D_{3.1}$-discs in G_1 is typically the same as that of $(\Psi(v_1), \ldots, \Psi(v_{s_{3.1}}))$ in G_2. This motivates the following definition of Φ acting on the first coordinate. Namely, abusing notation, for $S = (v_1, \ldots v_{s_{3.1}})$ we define $\Psi(S) = (\Psi(v_1), \ldots \Psi(v_{s_{3.1}})) \in \mathcal{Q}_2$ and define $\Phi(S, .) = (\Psi(S), .)$. Since Ψ is a bijection, it follows that for a uniformly distributed $S \in \mathcal{Q}_1$, the projection of Φ on its first coordinate induces the uniform distribution on $\mathcal{Q}_2$.

Now recall that the output of algorithm SAMPLER does not only depend on the $D_{3.1}$-discs of the sampled vertices, but also on the relative ordering that the permutation π induces on the sampled vertices. If we can define Φ in such a way, that not only the sampled vertices and their images have isomorphic $D_{3.1}$-discs, but also the relative ordering according to a random permutation π and its image under Φ is identical, we are done. Although, we won't be able to guarantee this for every pair $((S, \pi), \Phi(S, \pi))$, we will still succeed to show this for most pairs.

In order to define how Φ acts on the second coordinate, we will define for every $s_{3.1}$-sequence S a bijection $\phi_S : V_1 \to V_2$. This bijection is supposed to map, for every $1 \le i \le s_{3.1}$, the vertices inside the $D_{3.1}$-disc around sample vertex v_i in G_1 to the corresponding vertices in the corresponding $D_{3.1}$-discs around sample vertex $\Psi(v_i)$ in G_2. In particular, it is supposed to map v_i to $\Psi(v_i)$. However, a mapping as described above cannot always be done. Imagine, for example, the situation in which the $D_{3.1}$-discs $B_{G_1}(v_1, D_{3.1})$ and $B_{G_1}(v_2, D_{3.1})$ are disjoint in G_1 but the $D_{3.1}$-discs $B_{G_2}(\Psi(v_1), D_{3.1})$ and $B_{G_2}(\Psi(v_2), D_{3.1})$ are not disjoint in G_2. In this case, it may not be possible to extend certain orderings for the vertices in G_1 to an ordering of the vertices in G_2 such that the relative orderings among the vertices of $B_{G_1}(v_1, D_{3.1})$ and $B_{G_1}(v_2, D_{3.1})$ agree with that of $B_{G_2}(\Psi(v_1), D_{3.1})$ and $B_{G_2}(\Psi(v_2), D_{3.1})$. If for a given sample set and Ψ we cannot define a bijection ϕ_S with the desired properties, we will use an arbitrary bijection.

Let us continue now to define ϕ_S. Abusing notation, for a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ let us define $\phi_S(\pi) = (\phi_S(\pi_1), \phi_S(\pi_2), \ldots, \phi_S(\pi_n))$ to be the sequence that is obtained by identifying every vertex from π with its image under ϕ_S. Then we can define $\Phi(S, \pi) = \Phi(\Psi(S), \phi_S(\pi))$. Since, for every S, ϕ_S is a bijection, it follows that Φ is a bijection.

It remains to describe the construction of ϕ_S. Let us fix S. We first consider the situation, where

- the $D_{3.1}$-discs $\{B_{G_1}(v_i, D_{3.1}),\ i = 1, \ldots, s_{3.1}\}$ are pairwise disjoint,
- the $D_{3.1}$-discs $\{B_{G_2}(\Psi(v_i), D_{3.1}),\ \ i = 1, \ldots, s_{3.1}\}$, are pairwise disjoint, and
- the $D_{3.1}$-discs $B_{G_1}(v_i, D_{3.1})$ and $B_{G_2}(\Psi(v_i), D_{3.1})$ are isomorphic for each $1 \le i \le s_{3.1}$.

Let $\Gamma_S : \cup_{i=1}^{s_{3.1}} \{B_{G_1}(v_i, D_{3.1})\} \mapsto \cup_{i=1}^{s_{3.1}} \{B_{G_2}(\Psi(v_i), D)\}$ be an isomorphism that, for each i with $1 \le i \le s_{3.1}$, maps the root of the $D_{3.1}$-disc $B_{G_1}(v_i, D_{3.1})$ to the root of $B_{G_2}(\Psi(v_i), D_{3.1})$. We extend Γ_S arbitrarily to a bijection between V_1 and V_2. This extension will be our function ϕ_S. By construction, ϕ_S will identify every vertex in a $D_{3.1}$-disc $B_{G_1}(v_i, D_{3.1})$ with the corresponding vertex of the $D_{3.1}$-disc $B_{G_2}(\Psi(v_i), D_{3.1})$.

To finish our construction, let us consider the case when at least one of the three conditions above is not satisfied by S. In this case, we define ϕ_S to be an arbitrary bijection between V_1 and V_2. We will see below that this case happens only with small probability. Finally, $\Phi(S, \pi) = \Phi(\Psi(S), \phi_S(\pi))$.

CLAIM 6.3. *The map $\Phi : \mathcal{Q}_1 \times \Pi_1 \mapsto \mathcal{Q}_2 \times \Pi_2$ has the following properties.*

1. *The projection of Φ on its first coordinate is a bijection from $\mathcal{Q}_1$ to $\mathcal{Q}_2$, and the projection of Φ on its second coordinate, for any given S, is a bijection between Π_1 to Π_2.*
2. *For a pair (S, π) chosen uniformly at random from $\mathcal{Q}_1 \times \Pi_1$, the map $\Phi(S, \pi)$ induces a pair that is distributed uniformly on $\mathcal{Q}_2 \times \Pi_2$.*
3. *Let $(S, \pi) \in \mathcal{Q}_1 \times \Pi_1$ be a pair for which the $D_{3.1}$-discs around v_i, $i = 1, \ldots, s_{3.1}$, as well as the $D_{3.1}$-discs*

around $\Psi(v_i)$, $i = 1, \ldots, s_{3.1}$, are pairwise disjoint. Assume in addition that $B_{G_1}(v_i, D_{3.1})$ is isomorphic to $B_{G_2}(\Psi(v_i), D_{3.1})$, $i = 1, \ldots, s_{3.1}$, then there is an isomorphism ϕ_S between subgraphs $H_1 = \cup_{i=1}^{s_{3.1}} B_{G_1}(v_i, D_{3.1})$ and $H_2 = \cup_{i=1}^{s_{3.1}} B_{G_2}(\Psi(v_i), D_{3.1})$ that maps the roots of the $D_{3.1}$-discs in G_1 to the corresponding roots in G_2. In addition, the relative order on the vertices of H_1 that is defined by π is identical to the relative order on the vertices in H_2 that is defined by $\phi_S(\pi)$ under that isomorphism.

PROOF. The first and third items are by the definition of the construction. The second item follows directly from the first item. ■

With the bijection Φ between the probability spaces we define the coupling between the runs of the sampler for G_1 and G_2, naturally, in the following way: If, for graph G_1, the sampler chooses permutation π and samples a $s_{3.1}$-sequence S, then the randomness for the run on G_2 is $\Phi(S, \pi)$.

We next define the following five events on the probability space $\Pi_1 \times \mathcal{Q}_1$.

(A1) for all $1 \leq i < j \leq s_{3.1}$ the discs $B_{G_1}(v_i, D_{3.1})$ and $B_{G_1}(v_j, D_{3.1})$ are disjoint.

(A2) for all $1 \leq i < j \leq s_{3.1}$ the discs $B_{G_2}(\Psi(v_i), D_{3.1})$ and $B_{G_2}(\Psi(v_j), D_{3.1})$ are disjoint.

(A3) for all $1 \leq i \leq s_{3.1}$ the discs $B_{G_1}(v_i, D_{3.1})$ and $B_{G_2}(\Psi(v_i), D_{3.1})$ are isomorphic.

(B1) The two events asserted by Lemma 5.2, for the pair (ϵ, λ) in Lemma 5.2 set to $(\frac{\epsilon}{4}, \frac{\epsilon}{4})$ and k set to $k_{3.1}$, occur for G_1. Namely, the partition $\mathcal{P}_1 = \mathcal{P}(\pi)$ on G_1 is an $(\frac{\epsilon d}{4}, k_{3.1})$-partition, and the estimate $\tilde{f}_1$ computed by the sampler run on G_1 is $\frac{\epsilon}{4}$-close to f_1, where f_1 is the frequency vector of $G_1[\mathcal{P}_1]$ on $k_{3.1}$-discs.

(B2) The two events asserted by Lemma 5.2, for the pair (ϵ, λ) in Lemma 5.2 set to $(\frac{\epsilon}{4}, \frac{\epsilon}{4})$ and k set to $k_{3.1}$, occur for G_2. Namely, the partition $\mathcal{P}_2 = \mathcal{P}(\phi_S(\pi))$ on G_2 is an $(\frac{\epsilon d}{4}, k_{3.1})$-partition, and the estimate $\tilde{f}_2$ computed by the sampler run on G_2 is $\frac{\epsilon}{4}$-close to f_2, where f_2 is the frequency vector of $G_2[\mathcal{P}_2]$ on $k_{3.1}$-discs.

We will show in Claim 6.5 that with high probability all five events occur simultaneously. We then end the proof by the following Claim.

CLAIM 6.4. *If the events $A1, A2$ and $A3$ are satisfied, then the corresponding coupled runs of the Sampler on G_1 and G_2 return the same estimate vector $\tilde{f}$.*

PROOF. Let (S, π) be the random choice of the sampler of Lemma 5.2, for the pair $(\frac{\epsilon}{4}, \frac{\epsilon}{4})$ and input G_1. Let $(S', \pi') = \Phi((S, \pi))$ be the randomness for a run on input G_2. Let H_1 be the union of the discs $B_{G_1}(v_i, D_{3.1})$ for all $1 \leq i \leq s_{3.1}$ and let H_2 be the union of the discs $B_{G_2}(v'_i, D_{3.1})$ for all $1 \leq i \leq s_{3.1}$. The occurrence of events A1, A2 and A3 implies that H_1 and H_2 are isomorphic graphs and that there is an isomorphism that maps the roots of the corresponding $D_{3.1}$-discs to each other. Let this isomorphism be $\Gamma_S : V(H_1) \mapsto V(H_2)$. Further, by item 3. in Claim 6.3, the relative order induced by π on $V(H_1)$ is identical to the order that $\phi_S(\pi)$ induces on $V(H_2)$ under the map Γ_S. We conclude that the runs on G_1, G_2 see isomorphic views and so they must return the same vector (Note that it may happen with small probability that the discs are not large enough to determine the partition locally. In this case, we just need to assume that the runs behave similarly). ■

Assuming we have proved that all the events: $A1, A2, A3$ and , $B1, B2$ hold, we formally end the proof of the theorem, following the high level explanation, as follows.

Since all events hold simultaneously with positive probability, there is a pair in the probability space for which all events hold. Let (S, π) be such a pair. Since $A1, A2, A3$ occur, Claim 6.4 asserts that both runs compute the same estimate $\tilde{f}$. The fact that $B1$ holds for G_1 implies that the partition $\mathcal{P}(\pi)$ is an $(\frac{\epsilon d}{4}, k_{3.1})$-partition and that $G_1^* = G_1[\mathcal{P}(\pi)]$ has frequency vector of the connected components $f_1 = \mathrm{freq}_{G_1^*}(\mathrm{k}_{3.1})$ that is $\frac{\epsilon}{4}$-close to $\tilde{f}$. Similarly, the fact that the event $B2$ holds implies the same statement for $G_2^* = G_2[\mathcal{P}(\phi_S(\pi))]$ with a frequency vector f_2. Hence we conclude, by Claim 6.2 that G_1^* is $2\frac{\epsilon}{4}$-close to G_2^*. Finally, again by $B1, B2$, the partitions $\mathcal{P}(\pi)$ and $\mathcal{P}(\phi_S[\pi])$ are $(\frac{\epsilon d}{4}, k)$-partitions, which in particular implies that G_1 is $\frac{\epsilon}{4}$-close to G_1^* and G_2 is $\frac{\epsilon}{4}$-close to G_2^*. Combining this together with the fact that G_1^*, G_2^* are $\frac{\epsilon}{2}$-close implies that G_1 is ϵ-close to G_2.

CLAIM 6.5. *Let G_1, G_2, and Φ fixed as above. There is a function $= N(\epsilon, \rho, d)$ (independent of G_1, G_2 and Φ) such that for $n \geq N$ the events $A1, A2, A3$ and $B1, B2$ are simultaneously satisfied with probability at least $3/10$.*

PROOF. We show that each event holds with high enough probability and then use the union bound to conclude the assertion of the claim.

The probability that for a vertex v that is chosen uniformly at random from V_1, $B_{G_1}(v, D_{3.1})$ is not isomorphic to $B_{G_2}(\Psi(v), D_{3.1})$ is at most $\lambda_{3.1}$, by Claim 6.1 and our choice of $\lambda_{3.1}$. Therefore, the probability for event A3 not to hold is at most $\lambda_{3.1} \cdot s_{3.1} \leq 1/10$.

If regarded as stand-alone algorithms, i.e. without the coupling, both instances of the sampler of Lemma 5.2 sample $s_{3.1}$ vertices uniformly at random from the corresponding graphs. The value of $s_{3.1}$ as well as the maximum number of vertices in any $2D_{3.1}$-discs do not depend on n. Therefore, the probability that any two $D_{3.1}$-discs intersect goes to 0 as n goes to infinity. Hence, there is a value of $N = N(\epsilon, k, d)$ such that for all graphs with $n \geq N$ vertices, the probability of events $A1$ on G_1 and similarly $A2$ on G_2 not to occur is at most $1/10$.

Now, for each of G_1, G_2 separately, again, viewing the sampler run as a stand alone algorithm, Lemma 5.2 asserts that the sampler produces an estimate $\tilde{f}_i$, $i = 1, 2$ for a partition $\mathcal{P}(\pi)$ that is a an $(\frac{\epsilon d}{4}, k)$-partition, each with probability $4/5$. Hence, the events $B1$ and $B2$ each fail with probability $1/5$.

By summing up the failure probabilities the union bound yields that the overall failure probability is at most $7/10$ and so all events hold with probability at least $3/10$. ■

Choosing $N_{3.1} = N$ as in Claim 6.5 finishes the proof of Theorem 3.1. ■

7. PROOFS OF THEOREMS 3.2, 3.3 AND 3.4

Proof Of Theorem 3.2,. Let G_1, G_2 be two graphs on n vertices taken from a ρ-hyperfinite family, of bounded degree d. For $0 < \epsilon < 1$ let $\epsilon_{3.1} = \epsilon_{3.1}(\epsilon)$, $N = N_{3.1}(\epsilon, k, d)$, $D = D_{3.1}(\epsilon, k, d), \lambda = \lambda_{3.1}(\epsilon, k, d)$ from Theorem 3.1 and $k = \rho(\epsilon_{3.1})$. If $n < N$ then we can query all edges in G_1 and G_2 and solve the isomorphism problem exactly. So, let us assume that $n \geq N$. In this case, since G_1, G_2 are from a ρ-hyperfinite family of graphs, they are also $(\epsilon_{3.1}, k)$-hyperfinite. Hence, Theorem 3.1 can be applied. We can apply the estimator EstimateFrequencies of Section 5, with parameters D and $\lambda/4$, to output the frequency vectors $\widetilde{\mathrm{freq}_{\mathrm{G_i}}}(D)$, $i = 1, 2$. By Lemma 5.1, with probability at least $4/5$ both these vectors are $\lambda/4$-close to the real frequency vectors of the graphs G_1, G_2 respectively.

We then check if $|\widetilde{\mathrm{freq}_{\mathrm{G_1}}}(D) - \widetilde{\mathrm{freq}_{\mathrm{G_2}}}(D)|_1 \leq \lambda/2$. We accept if it is, and reject if the distance between the vectors is more than $\lambda/2$.

Suppose the graphs are isomorphic. Then obviously their corresponding frequency vectors are identical. Hence, with probability at least $8/10$ the distance we will observe between the estimates is at most $\lambda/2$ and the tester will accept.

Assume now that the graphs are ϵ-far from being isomorphic. We will show that the tester rejects with high probability. Indeed, assume that the tester accepts. This happens only if the distance between the estimates is at most $\lambda/2$, which implies that with probability at least $8/10$, the distance between the true frequency vectors is at most λ (by Lemma 5.1). But then, Theorem 3.1 implies that G_1 and G_2 are ϵ-close to isomorphic, contrary to the assumption. Thus the tester accepts with probability at most $2/10$. ■

Proof Of Theorem 3.3, **Sketch:**. Let $\mathcal{C}$ be ρ-hyperfinite family of d-bounded degree graphs. Let $0 < \epsilon < 1$ and $\epsilon_{3.1} = \epsilon_{3.1}(\epsilon)$, $N = N_{3.1}(\epsilon, k, d), D = D_{3.1}(\epsilon, k, d), \lambda = \lambda_{3.1}(\epsilon, k, d)$ from Theorem 3.1 and $k = \rho(\epsilon_{3.1})$. To define an ϵ-test for graphs of size n we define the following set (which might be infeasible to construct, but is well defined). Let $\mathcal{F}_{\mathcal{P}}(n, D) = \{\mathrm{freq_G(D)},\ \mathrm{G} \in \mathcal{C} \cap \mathcal{P},\ |\mathrm{V(G)}| = \mathrm{n}\}$ be the set of all possible frequency vectors of graphs of size n that are in $\mathcal{C}$ and have $\mathcal{P}$[6].

As in the previous proof it suffices to consider the case $n \geq N$ as in the other case we can query the whole graph. The test for a graph $G \in \mathcal{C}$, is to apply the estimator EstimateFrequencies of Section 5, with parameters D and $\lambda/2$. The estimator will output a frequency vector $\widetilde{\mathrm{freq}_{\mathrm{G}}}(D)$. By Lemma 5.1, with probability at least $9/10$, this vector is $\lambda/2$-close to the real frequency vector of G. We then check if $\widetilde{\mathrm{freq}_{\mathrm{G}}}(D)$ is $\lambda/2$-close to some vector in $\mathcal{F}_{\mathcal{P}}(n, D)$. We accept if it is and reject otherwise.

Obviously, if G has $\mathcal{P}$ then its frequency vector will be in $\mathcal{F}_{\mathcal{P}}(n, D)$ and since $\widetilde{\mathrm{freq}_{\mathrm{G}}}(\mathrm{D})$ is $\lambda/2$-close to $\mathrm{freq_G(D)}$ with probability at least $9/10$, the test will accept G (with that probability). On the other hand, if the test accepts G, then with probability at least $9/10$, $\mathrm{freq_G(D)}$ is λ-close to some vector $f \in \mathcal{F}_{\mathcal{P}}(n, D)$. By definition, this vector f is the frequency vector of some graph G' on n vertices in $\mathcal{P} \cap \mathcal{C}$, namely, $f = \mathrm{freq_{G'}(D)}$. But then, since the input graph is $(\epsilon_{3.1}, \rho(\epsilon_{3.1}))$-hyperfinite, Theorem 3.1 implies that G is ϵ-close to G'. This means that G is ϵ-close to $\mathcal{P}$ as needed. ■

Proof Of Theorem 3.4, **Sketch:**. The proof is very similar to the previous one. The only difference is that we now do not have the promise that G comes from a ρ-hyperfinite family of graphs. However, we know that the tested property is the intersection of some ρ-hyperfinite family with some other property. Hence, we can safely reject graphs, which are not ρ-hyperfinite. Our goal will be to accept with probability at least $9/10$ if the graph comes from a ρ-hyperfinite family of graphs and to reject with probability at least $9/10$, if the graph is not $(\epsilon_{3.1}, k)$-hyperfinite for a value of k determined below. A similar test has been used in [8] in their property testing algorithm for hereditary hyperfinite properties. For $k = \rho(\epsilon_{3.1}^3/(8 \cdot 54000))$, this is done by applying the local partitioning oracle using parameters $\epsilon_{3.1}/2$ and $\rho(\epsilon_{3.1}/2)$ and querying for random edges to estimate the number of edges between the partition classes. If this estimated number of edges is larger than $\frac{3}{4}\epsilon_{3.1}dn$, the algorithm rejects.

Thus, if the tester passes the first test, we know that it is $(\epsilon_{3.1}, k)$-hyperfinite, with high probability. So we can perform the test as in the proof of Theorem 3.3 and accept whenever this test accepts. The overall failure probability of the tester is at most $4/5$. ■

8. 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[7]. (b) properties that are defined by 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.

9. ACKNOWLEDGMENTS

We would like to thank the anonymous referees for the careful reading and helpful suggestions.

[6] instead of the whole set, we can take a constant size set that forms a δ'-net for $\mathcal{F}_{\mathcal{P}}(n, D)$. This does not change the non-uniformity nature, and the possible large time required to construct such a set. However, once it is constructed, it has only a constant size, hence the test for each n is given by a constant size table.

[7] needless to say, most if not all such previously constructed testers are much more efficient from these implied by Theorem 3.4 here

10. 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] 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.

[3] C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, B. Szegedy, Katalin Vesztergombi: Graph limits and parameter testing. STOC 2006: 261-270

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

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

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

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

[8] 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.

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

[10] 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.

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

[12] 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.

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

[14] 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).