

Testing H -Freeness on Sparse Graphs, the Case of Bounded Expansion

Samuel Humeau ✉ 

ENS de Lyon, CNRS, LIP, UMR 5668, 69342, Lyon cedex 07, France

Mamadou Moustapha Kanté ✉ 

Université Clermont Auvergne, Clermont Auvergne INP, LIMOS, CNRS, Clermont-Ferrand, France

Daniel Mock ✉ 

RWTH Aachen University, Germany

Timothé Picavet ✉ 

LaBRI, Université de Bordeaux, Talence, France

Alexandre Vigny ✉ 

Université Clermont Auvergne, Clermont Auvergne INP, LIMOS, CNRS, Clermont-Ferrand, France

Abstract

In property testing, a tester makes queries to (an oracle for) a graph and, on a graph having or being far from having a property P , it decides with high probability whether the graph satisfies P or not. Often, testers are restricted to a constant number of queries. While the graph properties for which there exists such a tester are somewhat well characterized in the dense graph model, it is not the case for sparse graphs. In this area, Czumaj and Sohler (FOCS'19) proved that H -freeness (i.e. the property of excluding the graph H as a subgraph) can be tested with constant queries on planar graphs as well as on graph classes excluding a minor.

Using results from the sparsity toolkit, we propose a simpler alternative to the proof of Czumaj and Sohler, for a statement generalized to the broader notion of bounded expansion. That is, we prove that for any class \mathcal{C} with bounded expansion and any graph H , testing H -freeness can be done with constant query complexity on any graph G in \mathcal{C} , where the constant depends on H and \mathcal{C} , but is independent of G .

While classes excluding a minor are prime examples of classes with bounded expansion, so are, for example, cubic graphs, graph classes with bounded maximum degree, or graphs of bounded book thickness. Additionally, random graphs with bounded average degree almost surely have bounded expansion.

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

Keywords and phrases Property testing, Sparsity, Bounded expansion, Treedepth

Digital Object Identifier 10.4230/LIPIcs.STACS.2026.55

Funding *Daniel Mock*: Supported by the Deutsche Forschungsgemeinschaft (DFG, German Science Foundation) – DFG-927/15-2.

Alexandre Vigny: This work benefited from state aid, managed by the ANR under France 2030 referenced ANR-23-IACL-0006.

1 Introduction

Given some data with a question, do you need to read all the data to answer, or can you parse only a fraction of it before answering correctly with high probability? The domain of study associated to such problems is known as *property testing*. Intuitively, a property testing algorithm (or *tester*) takes a graph G and a property P as inputs, accepts with high probability if G satisfies P , and rejects with high probability if G is “*far*” from having property P . The



© Samuel Humeau, Mamadou Moustapha Kanté, Daniel Mock, Timothé Picavet, and Alexandre Vigny;
licensed under Creative Commons License CC-BY 4.0

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).

Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thăng

Article No. 55; pp. 55:1–55:18



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



notion of being “far” from having a property is often described via the fraction ϵ of edge additions or edge removals needed to transform G into a graph G' satisfying P . The fixed fraction ϵ is often called the *proximity parameter*. A tester accepting with probability one on graphs satisfying P has *one-sided error*. Otherwise, the tester has *two-sided error*.

A tester does not have direct access to its input graph. Instead it uses an *oracle*, i.e. a “black box” serving as a representation of the input graph to which *queries* can be made. These depend on context and can be, for example: “are these two vertices neighbors?” or “give me a random neighbor of this vertex”.

There exists several models formalizing the notions described above: the dense, sparse, and bounded degree models, and they differ on how “being far from a property” is defined and on the available queries. The efficiency of testers is quantified by the number of queries, called *query complexity*, made to a given representation of G . Ideally, the number of queries is constant: it depends on the property P and the proximity parameter ϵ of “being far from P ”, but not on G . In this case, P is called *testable with constant query complexity*, or *testable*.

Formally, in the *dense graph model* introduced by Goldreich, Goldwasser, and Ron [12], a graph G is said to be ϵ -close to a property P if changing at most a fraction ϵ of its adjacency matrix (i.e. at most $\epsilon \cdot |V(G)|^2$ many edges) transforms G into a graph G' satisfying P . Otherwise G is called ϵ -far from P . For this model, there is a good understanding of what properties are testable with two-sided error [2], but one-sided errors as well [3, 4]. For example, every hereditary property (i.e. stable under taking induced subgraphs; such as bipartite, H -free or k -colorable) are one-sided error testable in the dense model.

In the bounded degree model [13], the input comes with an integer d and every vertex of the input is assumed to have degree at most d . In this case, it is known that restricting the class further to planar graphs, or graphs that exclude a minor (i.e. proper minor closed classes of graphs), and, more generally, to hyperfinite graph classes, allows every property to be testable with two-sided error [15]. For the more general case where only the degree constraint is considered, far fewer properties can be tested efficiently. On the one hand, it is known that testing FO-properties of the form $\exists\forall$ can be done efficiently. This includes testing for H -freeness. On the other hand some $\forall\exists$ -FO formulas cannot be tested efficiently [1]. One of the most important property that cannot be tested is bipartiteness, which requires (on general graphs with bounded degree) $\Omega(\sqrt{|G|})$ many queries. See for example chapter nine of [11].

The sparse model is currently the less known model. It splits into slightly different variations, depending on whether from a given vertex the algorithm can query: 1) its i th neighbor (upon input an integer i , and receives an error if the vertex has less than i neighbors), or 2) a random neighbor (possibly several times the same), or 3) a random distinct neighbor (distinct from the outputs of previous queries on the same vertex). We refer to the introduction of [9] for more information and, similarly to the recent work in this area [6, 7, 8, 10]. We focus here on the *random neighbor model*, i.e., the second variation of the sparse model as presented above. Observe that a query in this variation can be simulated using a constant number of queries in the other variations. Hence any property testable in the random neighbor model is testable in every variation of the sparse model. In a similar manner, a query from any variation of the sparse model can be simulated using a constant number of queries in the bounded degree model [9, Section 1.1.2]. Hence, it is natural to seek generalization of results from the sparse and bounded degree models, starting with the random neighbor model. Formal definitions for the latter are recalled in the preliminaries.

State of the art. The most impactful work in the sparse model is the tester for subgraph freeness on planar graphs (and actually all classes excluding a fixed minor) of Czumaj and Sohler [9]. Their algorithm simply repeats random breadth first searches with breadth and

depth independent of the input graph. There are several possible directions to generalize this work. First, providing algorithms testing broader properties. This is the case of [10], which uses the algorithm of [9] as a subroutine and shows how to test any monotone property in the sparse model.

Another direction is to look at more general graph classes, extending beyond minor-free graph classes. Minor-free graph classes do not contain all cubic graphs, so that the results in the bounded degree model are separated from the ones of [9]. So far, such an extension has only been achieved for testing the non-existence of cycles [6, 7] (on classes with bounded r -admissibility), and as Esperet and Norin noted in [10]: “*However, since the proof of [9] itself strongly relies on edge-contractions (and thus on the graph class \mathcal{C} being minor-closed), Theorem 2 [of [10]] does not seem to be easily extendable beyond minor-closed classes.*”

Furthermore, the proof of [9] is quite long and technical with key lemmas being dependent on their technical machinery. It is therefore hard to use corresponding insightful ideas individually. For examples, [10] uses Czumaj and Sohler’s tester as a black box, while [6, 7] do not seem to use any of their tools.

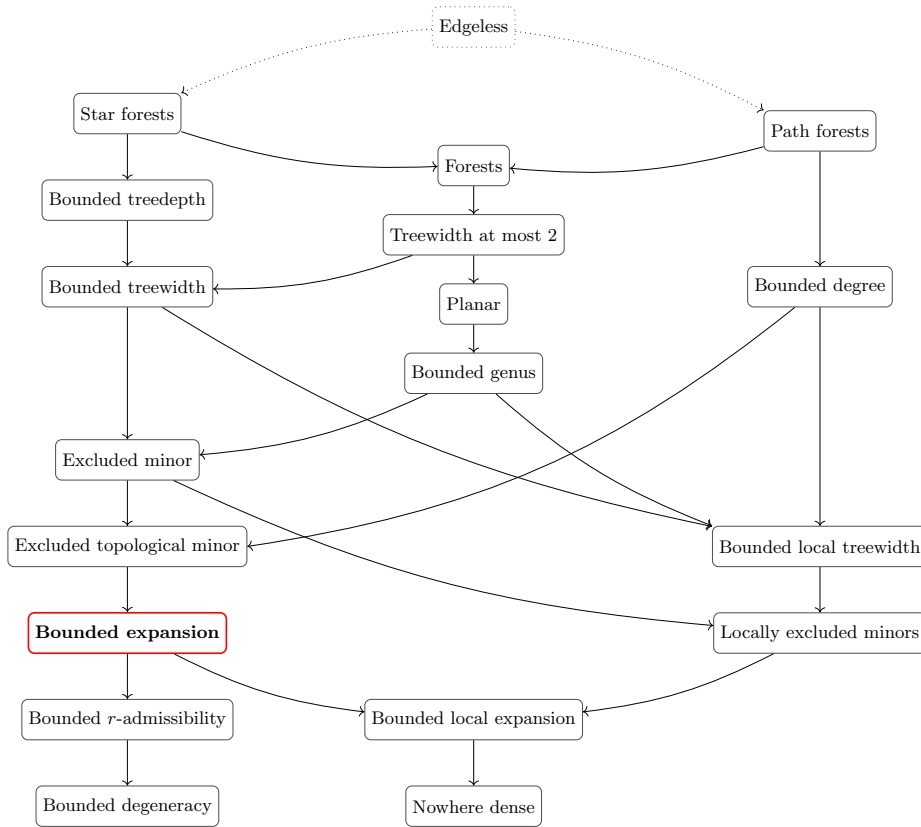
Our contribution. We prove that, for every proximity parameter, graph class \mathcal{C} of bounded expansion, and property P characterized by a finite set of forbidden subgraphs, P is testable on graphs of \mathcal{C} in the sparse model with constant query complexity and one-sided error. This result generalizes that of [9] from proper minor-closed graph classes to graph classes \mathcal{C} of bounded expansion. In particular, we get that such properties P are testable on graph classes of bounded degree and proper topologically minor-closed classes, generalizing results from the bounded degree model as well. A part of the hierarchy of sparse graph classes including these is represented in Figure 1.

We make sure to keep lemmas as independent of each other as possible, clearly stating what the requirements of each statement are. While some lemmas work on graphs from a bounded expansion class, some only require their degeneracy to be bounded, and some hold for arbitrary graphs. Lemmas and definitions which are reformulations or slight variations of statements of [9] are explicitly pointed out as such.

Independently, Awefeso et al. discovered the same result which is now available as a preprint [5]. Indeed, they are more precise on which graph classes the H -freeness property is testable; for graphs H of size k , it is sufficient that the graph class have so-called bounded k -admissibility. These are family of notions which all subsume the notion of bounded expansion and are subsumed by the notion of bounded degeneracy. Note that in order to test H -freeness for arbitrary H , their proof requires bounded k -admissibility for arbitrary k , which then coincides with bounded expansion.

Organization and techniques. To prove our result, we first proceed in the preliminaries in the same manner as Czumaj and Sohler [9]: we focus on the testability of H -freeness for a fixed graph, and we use the same tester: repeated bounded random breadth first searches accepting if a copy of H is found by the searches, and rejecting otherwise. One-sidedness is direct, so most of the proof is to handle graphs which are ϵ -far from being H -free. As pointed out by Czumaj and Sohler, such a graph G contain $\Omega(\epsilon|V(G)|)$ edge-disjoint copies of H .

Hence, during our proof procedure, we handle and maintain a “sufficiently large” set \mathcal{H} of edge-disjoint copies of H in G . With each lemma, we restrict \mathcal{H} to a subset of itself, imposing further structure on the subgraph $G[\mathcal{H}]$ induced by the copies of H in \mathcal{H} . In Section 3.1, we study a property on \mathcal{H} enabling to transfer the probability of finding H in $G[\mathcal{H}]$ to that of finding H in G . Later in Section 3.2, we explain how to build a set \mathcal{H} such that $G[\mathcal{H}]$ has bounded treedepth. Using that having bounded treedepth implies excluding a minor, together with the result of [9], we prove our main contribution in Section 3.3: Theorem 3.8.



■ **Figure 1** Families of sparse graph classes and their pairwise inclusions.

For the sake of completeness, and in order not to rely on [9], we show how to test for H -freeness on graphs with bounded treedepth in Section 4. This is done by first analyzing the structure of H with regard to tree embeddings of bounded depth, and later, by analyzing how parts of edge-disjoint copies of H can be used to reconstruct a copy of H in a graph of bounded treedepth. With Theorem 4.17, we finally conclude that H -freeness can be tested with constant query complexity and one-sided error in the random neighbor model on graphs of bounded treedepth.

2 Preliminaries

2.1 Graph notations

We denote the vertex and edge set of a graph G by $V(G)$ and $E(G)$, respectively. We write $|G|$ for the number of vertices of G . We assume that graphs are finite, simple and undirected.

Given a graph G and a set of vertices A of G , we denote by $G[A]$ the *subgraph* of G induced by the vertices in A (i.e. keeping every edge incident to two vertices of A). For a set F of edges of a graph G , we denote by $G - F$ the subgraph of G with vertex set $V(G)$ and edge set $E(G) \setminus F$.

The *degree* $\deg^G(v)$ of a vertex v in a graph G is the number of vertices v shares an edge with in G . The *degree* $\Delta(G)$ of a graph G is the maximum degree of its vertices.

► **Definition 2.1.** A copy h of a graph H in a graph G is a subgraph isomorphism, i.e. an injection $h: V(H) \rightarrow V(G)$ such that $(u, v) \in E(H) \implies (h(u), h(v)) \in E(G)$ for every $u, v \in V(H)$.

Given a set \mathcal{H} of copies of H in G , $G[\mathcal{H}]$ denotes the subgraph of G defined by the union of the subgraphs of G image of elements of \mathcal{H} .

A graph G is said to be H -free if it does not contain H as a subgraph, i.e., there are no copies of H in G .

2.2 Sparsity

We start with the definition of treedepth, and then move on to the more general notion of bounded expansion introduced by Nešetřil and Ossona de Mendez. We refer to their more comprehensive book on the subject for more information [16].

A tree order \leq_T on a vertex set V is a partial order such that for every vertex v , the set of elements \leq_T -smaller than v is totally ordered. A root with regard to \leq_T is a \leq_T -minimal element, and a leaf with regard to \leq_T is a \leq_T -maximal element. The level of a vertex v in a tree order is the number of elements smaller than v . The depth of a tree order is the maximal level of its vertices. In particular, roots have level zero. A tree order \leq_T on the vertex set of a graph is called a tree embedding (or treedepth embedding) if every pair of adjacent vertices (u, v) is \leq_T -comparable (i.e. either $u \leq_T v$, or $v \leq_T u$).

► **Definition 2.2.** The treedepth $\text{td}(G)$ of a graph G is the minimum depth of its tree embeddings.

For an integer d , a graph class \mathcal{C} is said to have treedepth at most d if the treedepth of each of its graphs is bounded by d ; \mathcal{C} has bounded treedepth if there exists such a constant d .

The notion of bounded expansion was introduced by Nešetřil and Ossona de Mendez [14]. It has many equivalent characterizations. As our proofs rely only on the existence of low-treedepth colorings, we only recall the corresponding characterization.

A p -treedepth coloring of a graph G is a coloring of the vertices of G such that for every set S of $i \leq p$ colors, the subgraph of G induced by its vertices colored by elements of S has treedepth at most i .

► **Definition 2.3** (See Theorem 7.1 of [14]). A class of graphs \mathcal{C} has bounded expansion if and only if there exists a function f such that for every graph $G \in \mathcal{C}$ and all $p \in \mathbb{N}$, G admits a p -treedepth coloring using at most $f(p)$ colors.

Note that, while such colorings can be computed efficiently, our proof only uses their existence. There are many variations on such coloring, often grouped in the notion of generalized coloring numbers. This includes among other weak-coloring, p -centered coloring, and admissibility coloring. The latter is used in [6, 7] as (r -) admissibility numbers. In our case, we sometimes use the notion of degeneracy, which is equivalent to the notion of 1-admissibility.

► **Definition 2.4.** A graph has degeneracy d if every subgraph has a vertex of degree at most d . A class of graphs has bounded degeneracy if there exists a constant d such that every graph in the class has degeneracy at most d .

Classes of bounded expansion are well-known to have bounded degeneracy [14, Fact 3.2].

2.3 Property testing and complexity

For a positive constant $\epsilon > 0$, a graph G is ϵ -far from a property P if more than $\epsilon \cdot |G|$ many edges have to be deleted from or inserted to G to obtain a graph with property P . The parameter ϵ is called the *proximity parameter*.

A *tester* (for a property P) is an algorithm that inspects only a part of an input graph G via a restricted set of *queries* to an *oracle*, accepts with probability at least $2/3$ if G satisfies P , and rejects with probability at least $2/3$ if G is ϵ -far from P . A tester has *one-sided error* if it accepts all graphs with property P with probability 1.

In [9], several oracle access models are presented, but the focus is put on the random neighbor model. We proceed similarly.

Initially, a tester knows the vertices of its input graph G ; it accesses the input graph by making *random neighbor queries* to its oracle: given a vertex $v \in V(G)$, the oracle returns a vertex chosen independently and uniformly at random from the set of all neighbors of v . No other queries are available to testers.

The *query complexity* of a tester is the number of queries it makes before reaching a conclusion.

In our work, we only consider *constant query complexities*: the number of queries is always independent of the input graph. More precisely, when testing for H -freeness with proximity parameter ϵ , the number q of queries can depend on H , ϵ , and additional parameters (such as a fixed class \mathcal{C} the graph G belongs to). Hence, $q = O_{H,\epsilon,\mathcal{C}}(1)$, that is, for some function $f(H,\epsilon,\mathcal{C})$, we have $q \leq f(H,\epsilon,\mathcal{C})$. This notation extends easily, by ignoring multiplicative factors depending on indices; for example stating $|\mathcal{H}| = \Omega_{\epsilon,H}(|V(G)|)$ for a set of copies of H in G .

2.4 Random Bounded BFS and the tester

First we show that testing for \mathbb{H} -freeness with \mathbb{H} a set of graphs is equivalent to testing for H -freeness for a single graph H . This is identical to a part of [9, Section 8]. We only prove it for one-sided error testers since this is what we later investigate.

First, we relate ϵ -farness from being \mathbb{H} -free or H -free for H some graph in \mathbb{H} :

► **Proposition 2.5.** *Let G be a graph and \mathbb{H} a finite set of graphs.*

If G is ϵ -far from being \mathbb{H} -free, then there exists a graph $H \in \mathbb{H}$ such that G is $\epsilon/|\mathbb{H}|$ -far from being H -free.

Proof. For the sake of contradiction, assume the statement is false. Hence, for each graph H of \mathbb{H} , it is possible to remove at most $\epsilon|G|/|\mathbb{H}|$ edges from G to make it H -free. If we do this for all graphs of \mathbb{H} , then we remove at most $\epsilon|G|$ edges from G and make it \mathbb{H} -free, a contradiction. ◀

► **Proposition 2.6.** *Let \mathbb{H} be a finite set of graphs and \mathcal{C} a graph class.*

If, for every graph H of \mathbb{H} and $\epsilon > 0$, H -freeness is testable with constant query complexity, proximity parameter ϵ , and one-sided error in the random neighbor model on graphs of \mathcal{C} , then so is \mathbb{H} -freeness.

Proof. Given G, ϵ, \mathbb{H} , consider the algorithm iteratively testing H -freeness (with proximity parameter $\epsilon/|\mathbb{H}|$) for each $H \in \mathbb{H}$. If the sub-tester for one of the graph $H \in \mathbb{H}$ rejects, then the algorithm rejects G . Otherwise, the algorithm accepts G . The number of queries is constant (depending only on ϵ and \mathbb{H} , and possibly \mathcal{C}).

Note that if G is indeed \mathbb{H} -free, since sub-testers are one-sided, they all accept, and therefore our algorithm accepts. Otherwise, if G is ϵ -far from \mathbb{H} -freeness, by Proposition 2.6, there is exists $H \in \mathbb{H}$ such that G is $\epsilon/|\mathbb{H}|$ -far from being H -free. With probability at least $2/3$ the sub-tester for H rejects, and therefore our algorithm rejects with probability at least $2/3$ too. \blacktriangleleft

Our tester for H -freeness is relatively straightforward. It is the same as that of [9], except that we provide formal details on how disconnected graphs are handled, whereas this particular case is only discussed informally in [9, Section 8]. Our tester is built on the subroutine for Random Bounded Breadth First Searches: `RandomBoundedBFS` described below.

■ **Algorithm 1** `RandomBoundedBFS(G, t, d)`.

Input: Graph G , integers t, d

- 1 Pick a vertex v uniformly at random in G ;
- 2 `current_vertices` \leftarrow $\{v\}$; `final_edges` \leftarrow \emptyset ;
- 3 `next_vertices` \leftarrow \emptyset ; `seen_vertices` \leftarrow \emptyset ;
- 4 **for** $i = 1$ **to** t **do**
- 5 **foreach** $u \in$ `current_vertices` **do**
- 6 perform d random neighbor queries on u ;
- 7 **foreach** *edge* uw *found by a query* **do**
- 8 **if** $w \notin (\text{seen_vertices} \cup \text{current_vertices})$ **then**
- 9 `add` w `to` `next_vertices`;
- 10 `add` uw `to` `final_edges`;
- 11 `add` u `to` `seen_vertices`;
- 12 `current_vertices` \leftarrow `next_vertices`; `next_vertices` \leftarrow \emptyset ;
- 13 **return** The subgraph of G induced by edges in `final_edges`;

Hence, `RandomBoundedBFS` does a breadth first search, with bounded breadth and depth, and at random. Our final algorithm `Tester(n, H, ϵ, G)` rejects when $|H| > |G|$. Otherwise it removes its isolated vertices from H and repeats the following n times (where n should be independent of G to ensure constant query complexity):

1. for every connected component C of H , make a call to `RandomBoundedBFS($G, |H|, \Delta(H)$)`;
2. if the union of the subgraphs returned by the calls to `RandomBoundedBFS` contain a copy of H , then reject. If at the end of the n iterations the algorithm has not rejected G , then accept.

Observe that a call to `RandomBoundedBFS` associated to a connected component C of H does not need to find a copy of H in the algorithm. Having one call per component ensures high probability. Concerning isolated vertices of H , once a copy of the non-isolated parts of H is found, any other vertex of G can be used as a copy of an isolated vertex of H . And since we have $|G| \geq |H|$ there are such vertices.

3 Testing H -freeness on graphs of bounded expansion

In this section, we prove our main result, Theorem 3.8: H -freeness can be tested in the random neighbor model with constant query complexity and one-sided error on a class of graphs of bounded expansion for any fixed graph H .

The proof is divided into multiple lemmas which maintain a set \mathcal{H} of copies of H in G . Instead of analyzing the success probability of the tester on G , we analyze it on $G[\mathcal{H}]$. With each lemma, the set \mathcal{H} is refined to obtain more properties. While \mathcal{H} shrinks with each lemma, we ensure that it remains “sufficiently large” – so the success probability of the tester on $G[\mathcal{H}]$ lifts to G . Essentially, the size of \mathcal{H} is always linear in the graph size $|G|$, where multiplicative factors depending only on ϵ , $|H|$ and graph structure parameters are considered constant. At the end of this process, we obtain a set \mathcal{H} such that $G[\mathcal{H}]$ has bounded treedepth. The main result follows then from [9], as classes of bounded treedepth are properly minor-closed.

However, our work does not end there. We give a self-contained, simple-to-follow proof that H -freeness is testable on classes of bounded treedepth in Section 4.

3.1 Reduction to subgraphs induced by many edge-disjoint copies of H

In this section, we show that any graph G that is ϵ -far from being H -free, has an associated set \mathcal{H} of copies of H with basic properties ensuring we can analyze the behavior of the tester on the structurally better-behaved graph $G[\mathcal{H}]$ instead of G . More precisely, the set \mathcal{H} has the following properties:

- the copies in \mathcal{H} are edge-disjoint,
- the number of copies is “sufficiently large”, that is, linear in $|G|$, and,
- the degrees of a vertex of $G[\mathcal{H}]$ in $G[\mathcal{H}]$ and G are of the same order of magnitude.

The findings in this section are generalizations or distillations of results from [9] into our setting.

► **Lemma 3.1.** *Let H and G be graphs such that G is ϵ -far from being H -free.*

There exists a set \mathcal{H} of edge-disjoint copies of H in G of cardinality at least $\epsilon|G|/|E(H)|$.

This lemma is a reformulation of a part of [9, Lemma 20] and is true on arbitrary graphs.

Proof. We prove, given a set \mathcal{H} of $k < \epsilon|G|/|E(H)|$ edge-disjoint copies of H in G , that we can build a set of $k + 1$ edge-disjoint copies of H in G .

Removing all edges of $G[\mathcal{H}]$ from G removes $k|E(H)| < \epsilon|G|$ edges from G . Since G is ϵ -far from being H -free, at least one copy of H remains in $G - E[G[\mathcal{H}]]$. Adding it to \mathcal{H} provides a set of $k + 1$ edge-disjoint copies of H in G . ◀

We define a condition on subgraphs of G and prove that the probability of finding H on a subgraph of G satisfying this condition lifts to G , up to a constant factor. This condition relates to [9, Property (a’), defined in Lemma 47].

► **Definition 3.2.** *Let $0 < c \leq 1$ be a constant. A vertex v in a subgraph G' of G is called c -degree preserved in G' if $\deg^{G'}(v) \geq c \cdot \deg^G(v)$. If all vertices of G' are c -degree preserved, and $|G'| \geq c|G|$, then G' is called c -degree preserving.*

► **Lemma 3.3.** *Let G be a graph, $0 < c \leq 1$ a positive constant, and G' a c -degree preserving subgraph of G . If $\text{RandomBoundedBFS}(G', d, t)$ returns a subgraph F of G' with probability at least q , then $\text{RandomBoundedBFS}(G, d, t)$ also returns F with probability at least $c^{d^t+1}q$.*

Again, this lemma is similar to [9, Lemma 47], the main difference being that in [9], the authors assume G' to be a spanning subgraph instead of having $|G'| \geq c|G|$ in the definition of c -degree preserving subgraph.

Proof. Let F be a subgraph of G' which is returned by $\text{RandomBoundedBFS}(G', d, t)$ with probability q . Assume the search starts on a vertex v of G' .

Since G' is c -degree preserving, $\text{RandomBoundedBFS}(G, d, t)$ now starts on v with probability $1/|G| \geq c/|G'|$ (this is the $+1$ in c^{d^t+1}).

Let uw be an edge of G' . Assume, during runs of $\text{RandomBoundedBFS}(G', d, t)$ and $\text{RandomBoundedBFS}(G, d, t)$ both starting on v , that u has just been visited, and uw has not been discovered yet by either runs.

Since G' is c -degree preserving, the neighborhood of u in G is larger than in G' by a factor of at most $1/c$. The probabilities that the runs of $\text{RandomBoundedBFS}(G', d, t)$ and $\text{RandomBoundedBFS}(G, d, t)$, just after visiting u , discover uw are, respectively:

$$\frac{1}{\deg^{G'}(u)} \quad \text{and} \quad \frac{1}{\deg^G(u)} \geq \frac{c}{\deg^{G'}(u)}.$$

In other words, the probability that the run of $\text{RandomBoundedBFS}(G, d, t)$ discovers uw just after visiting u is at least c times the probability that the run of $\text{RandomBoundedBFS}(G', d, t)$ discovers uw just after visiting u .

By construction, there are at most d^t edges in F : the probability that $\text{RandomBoundedBFS}(G, d, t)$ discovers precisely the edges of F , and in the same order than $\text{RandomBoundedBFS}(G', d, t)$ does, is at least $c^{d^t}q$.

Combining this with the probability that $\text{RandomBoundedBFS}(G, d, t)$ starts on v , the probability that $\text{RandomBoundedBFS}(G, d, t)$ finds a copy of F in G is at least $c^{d^t+1}q$. ◀

Finally we prove that degenerate graphs containing a set \mathcal{H} of many edge-disjoint copies of H also contain a “significantly large” set $\mathcal{H}' \subseteq \mathcal{H}$ such that $G[\mathcal{H}']$ is c -degree preserving.

► **Lemma 3.4.** *Let H be a graph with at least one edge and G a d -degenerate graph for a constant $d > 0$. Furthermore, let \mathcal{H} be a set of edge-disjoint copies of H in G . If $|\mathcal{H}| \geq \alpha|G|$ for a positive number $\alpha > 0$, then there exists $\mathcal{H}' \subseteq \mathcal{H}$ such that $G[\mathcal{H}']$ is $(\alpha/4d)$ -degree preserving and $|\mathcal{H}'| \geq \alpha|G|/2$.*

This lemma generalizes [9, Lemma 18] to graphs with bounded degeneracy. The assumption that $|E(H)| \geq 1$ (at the end of the proof) is not a limitation with regard to our tester, since **Tester** treats isolated vertices separately from the rest of H .

Proof. We prove this statement by providing a simple algorithm computing such a set \mathcal{H}' : take $\mathcal{H}' = \mathcal{H}$ at first. While there exists a vertex of $G[\mathcal{H}']$ which is not $(\alpha/4d)$ -degree preserving in $G[\mathcal{H}']$, choose one, say v , and remove from \mathcal{H}' every copy of H containing v in their image. This procedure starts with $\mathcal{H}' = \mathcal{H}$ and ends when all vertices in $G[\mathcal{H}']$ are $(\alpha/4d)$ -degree preserved. It clearly terminates after a finite number of steps.

In the following, we first argue that $|\mathcal{H}'| \geq \alpha|G|/2$ and then that $|G[\mathcal{H}']| \geq \alpha|G|/4d$.

Let v be a vertex that is chosen by the algorithm to remove all its incident copies. Once the vertex v is chosen by the algorithm, it cannot be selected again. Since the degree of v at the moment of its selection is bounded by $\alpha \deg^G(v)/4d$, and as the copies in \mathcal{H} are edge-disjoint, the number of copies removed in this step is at most $\alpha \deg^G(v)/4d$.

We bound the total number of copies removed during the algorithm by summing the bounds over *all* vertices v of G :

$$|\mathcal{H}| - |\mathcal{H}'| \leq \sum_{v \in G} \frac{\alpha \deg^G(v)}{4d} = \frac{\alpha|E(G)|}{2d}.$$

55:10 Testing H -Freeness on Sparse Graphs, the Case of Bounded Expansion

Since, furthermore, $|E(G)| \leq d|G|$ and $|\mathcal{H}| \geq \alpha|G|$:

$$|\mathcal{H}'| \geq |\mathcal{H}| - \frac{\alpha|E(G)|}{2d} \geq \alpha|G| - \frac{\alpha|E(G)|}{2d} \geq \alpha|G| - \frac{\alpha|G|}{2} = \frac{\alpha|G|}{2}.$$

Since $G[\mathcal{H}']$ is a subgraph of G , $G[\mathcal{H}']$ is also d -degenerate, and we have

$$|G[\mathcal{H}']| \geq \frac{|E(G[\mathcal{H}'])|}{d} \geq \frac{\alpha|G| |E(H)|}{2d} \geq \frac{\alpha|G|}{4d},$$

since $|E(H)| \geq 1$. Therefore, \mathcal{H}' is indeed $(\alpha/4d)$ -degree preserving. \blacktriangleleft

► **Corollary 3.5.** *Let H be a graph containing n_H connected components, and let G be a d -degenerate graph (for some constant $d > 0$) that is ϵ -far from being H -free.*

There exists a set \mathcal{H} of at least $\epsilon|G|/2|E(H)|$ edge-disjoint copies of H in G such that if, for a positive integer $n > 0$, $\text{Tester}(n, H, \epsilon, G[\mathcal{H}])$ finds a copy of H in $G[\mathcal{H}]$ with probability at least $2/3$, then there exists a positive integer $n' = O_{\epsilon, H, n}(1)$ such that $\text{Tester}(n', H, \epsilon, G)$ does so in G with probability at least $2/3$ in G .

Proof. We define \mathcal{H} with Lemmas 3.1 and 3.4. By Lemma 3.3, $\text{Tester}(n, H, \epsilon, G)$ finds the exact same subgraph of $G[\mathcal{H}]$ as $\text{Tester}(n, H, \epsilon, G[\mathcal{H}])$ with probability

$$p = (\epsilon/4d|E(H)|)^{nn_H(\Delta(H)^{|H|+1})}.$$

(Calls to `RandomBoundedBFS` being independent.) Since $\text{Tester}(n, H, \epsilon, G[\mathcal{H}])$ finds H with probability at least $2/3$, this implies that $\text{Tester}(n, H, \epsilon, G)$ finds H with probability at least $p' = 2p/3$. Note that, given a positive integer $m \geq 1$, running m times $\text{Tester}(n, H, \epsilon, G)$ is the same as running $\text{Tester}(nm, H, \epsilon, G)$. The probability that $\text{Tester}(n, H, \epsilon, G)$ fails m times in a row is at most $(1 - p')^m$.

Take $m = \lceil 2/p' \rceil$ and $n' = nm$, we conclude that $\text{Tester}(n', H, \epsilon, G)$ succeeds with probability at least $1 - (1 - p')^m = 1 - (1 - 2/m)^m > 1 - e^{-2} > 2/3$. \blacktriangleleft

3.2 Reduction to the case of graphs with bounded treedepth

We first prove that in a graph G which is ϵ -far from being H -free and from a class of bounded expansion, we can find a set \mathcal{H} of many edge-disjoint copies of H such that $G[\mathcal{H}]$ has bounded treedepth. The proof relies on the existence of p -treedepth colorings. We then reduce the problem of testing H -freeness on graphs of bounded expansion to graphs of bounded treedepth, by combining this lemma with the results from the previous section.

► **Lemma 3.6.** *Let H and G be graphs, with G from a graph class \mathcal{C} of bounded expansion. Let \mathcal{H} be a set of edge-disjoint copies of H in G .*

There exists $\mathcal{H}' \subseteq \mathcal{H}$ such that $|\mathcal{H}'| = \Theta_{|H|, \mathcal{C}}(|\mathcal{H}|)$ and $G[\mathcal{H}']$ has treedepth at most $|H|$.

Proof. Consider a $|H|$ -treedepth coloring of $G[\mathcal{H}]$, using colors c_1, \dots, c_ℓ . By Definition 2.3, ℓ is bounded by a function only depending on \mathcal{C} and $|H|$. Let $\{C_1, \dots, C_{\ell^{|H|}}\}$ be the set of all $|H|$ -tuples of colors. Each copy of H in \mathcal{H} contains $|H|$ vertices, and therefore is colored with the colors of at least one of the C_i . Therefore, by the pigeonhole principle, there is a tuple of colors C_i such that at least $|\mathcal{H}|/\ell^{|H|}$ copies of H in \mathcal{H} are colored with the colors of C_i . Let \mathcal{H}' be the set of copies that only use the colors of C_i .

Notice that the graph $G[\mathcal{H}']$ is a subgraph of $G[C_i]$, that is, the subgraph induced by the vertices colored with colors from C_i in the $|H|$ -treedepth coloring. By definition of the $|H|$ -treedepth coloring, $G[C_i]$ (and therefore $G[\mathcal{H}']$) has treedepth $|H|$. We can then conclude using $|\mathcal{H}'|/\ell^{|H|} \leq |\mathcal{H}'| \leq |\mathcal{H}|$, so $|\mathcal{H}'| = \Theta_{|H|, \mathcal{C}}(|\mathcal{H}|)$. \blacktriangleleft

We can now prove the reduction from bounded expansion to bounded treedepth for the success probability of the tester.

► **Lemma 3.7.** *Let H and G be graphs, with G ϵ -far from being H -free and from a graph class \mathcal{C} of bounded expansion.*

There exists a set \mathcal{H} of edge-disjoint copies of H in G such that $G[\mathcal{H}]$ has treedepth at most $|H|$ and, given a positive integer $n > 0$, if $\text{Tester}(n, H, \epsilon, G[\mathcal{H}])$ finds a copy of H in $G[\mathcal{H}]$ with probability at least $2/3$, then there exists a positive integer $n' = O_{\epsilon, H, n}(1)$ such that $\text{Tester}(n', H, \epsilon, G)$ finds a copy of H in G with probability at least $2/3$.

Proof. We start with Lemma 3.1 which gives a set \mathcal{H}_1 of $\Omega_{\epsilon, H}(|G|)$ many edge-disjoint copies of H . We then apply Lemma 3.6 yielding a set \mathcal{H}_2 such that $G[\mathcal{H}_2]$ has treedepth at most $|H|$ and $\mathcal{H}_2 = \Omega_{\epsilon, H, \mathcal{C}}(|G|)$. Finally, we apply Lemma 3.4 to extract a set \mathcal{H}_3 from \mathcal{H}_2 that is c -degree preserving for some constant $c = O_{\epsilon, H, \mathcal{C}}(1)$, and still $\mathcal{H}_3 = \Omega_{\epsilon, H, \mathcal{C}}(|G|)$.

We conclude similarly as for Corollary 3.5: thanks to Lemma 3.3. ◀

3.3 Testing H -freeness on graphs of bounded expansion

We are ready to prove the main result of this work.

► **Theorem 3.8.** *For any graph class \mathcal{C} of bounded expansion, any graph H , and any proximity parameter, the property of being H -free can be tested in the random neighbor model on graphs of \mathcal{C} with constant query complexity and one-sided error.*

Proof. By Lemma 3.7, it is sufficient to prove the statement when \mathcal{C} is a class of graphs with bounded treedepth. This is a particular case of the main result of [9], since bounded treedepth classes are properly minor-closed. ◀

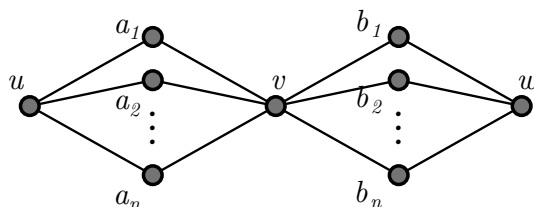
In this proof, we have relied on [9] for graphs of bounded treedepth. However, we reprove this particular case in the next section. Hence we present a self-contained proof, that we believe to be simpler since it avoids the machinery of edge-contractions, hypergraphs and safety from [9]. Our proof also provides an insight into the structure imposed by H -freeness on graphs of bounded treedepth.

We believe that classes of bounded treedepth offer rich structure for property testing and thus are a suitable “simple” class in the lower ends of the sparsity hierarchy (see Figure 1). Most of the existing literature on property testing in the sparse model focuses on graphs of bounded degree or on planar graphs and does not consider classes of bounded treedepth¹.

4 Testing H -freeness for graphs with bounded treedepth

We start by explaining how we proceed before giving the formal definitions of the terms used in the following introduction.

Consider the following family of graphs (a graph for each $n > 0$, similar to [8, Figure 3]):



¹ A notable exception is a work by Esperet and Norin [10] considering treedepth to prove 1. a linear Erdős–Pósa property for monotone properties on properly minor-closed classes and 2. that proper minor-closed classes admit approximate proof labelling schemes of logarithmic complexity.

The maximum length of its paths is five, and its treedepth at most five as well. Furthermore, it contains n edge-disjoint copies of P_5 of the shape ua_ivb_jw . Write \mathcal{H} for the corresponding set of copies of P_5 . Consider a run of `RandomBoundedBFS`. When $n \rightarrow \infty$, the probability of finding a copy of P_5 which is in \mathcal{H} goes to zero. However, with breadth one and depth four, `RandomBoundedBFS` finds a copy of P_5 of shape ua_ivb_jw with high probability, independently of the value of n .

`RandomBoundedBFS` also finds other kinds of copies of P_5 with high probabilities: $a_ivb_jwb_k$ for example. However, we focus on the copies of P_5 which are related to those in \mathcal{H} , i.e. that agree on common vertices (the vertices u , v , and w): each vertex of G corresponds to exactly one vertex of H via copies in \mathcal{H} . We say that G is *uniformly colored* by the vertices of H .

The particularity of copies of P_5 of shape ua_ivb_jw is that they consist in reconstructed copies of H from *parallel parts* of H (reduced here to the vertices a_i and b_j) attached to a few number of vertices (the vertices u , v , and w) we call *sources*. The specific choices of *parallel parts* does not matter as long as some *compatibility* is ensured on the sources.

In this section, we show that this behavior is not specific to copies of P_5 in the family of graphs introduced above, and we extend these ideas to graphs of bounded treedepth.

At first, we introduce formally the notions of *uniformly colored* set of copies of H . Furthermore, we relate the copies of H to the treedepth via *uniformly layered* treedepth embedding. We show that testing on graphs of bounded treedepth can be reduced to testing on graphs of treedepth at most $|H|$ with uniformly layered embeddings.

Then, we formalize the notions of parallel parts, sources, compatibility of parts, and prove that in a graph of treedepth at most $|H|$ with a uniformly layered embedding, copies of H can be constructed somewhat blindly using parallel parts, by choosing via which copy of a source we enter the next parallel part.

At last, we show that such constructions of connected components of H can be simulated in the `RandomBoundedBFS` assuming a breadth of $\Delta(H)$ and a depth of $|H|$, and with probability $\Omega_H(1)$. As a consequence, we conclude that H -freeness can be tested in the random neighbor model with constant query complexity and one-sided error on classes of graphs of bounded treedepth in the case H is connected. Finally, we extend this result to nonnecessarily connected graphs H by using uniformly layered embedding. Indeed, such embeddings ensure that with high probability, we find disjoint copies of the connected components of H during successive runs of `RandomBoundedBFS`.

4.1 Reduction to uniformly layered treedepth embeddings

► **Definition 4.1.** *Let H and G be two graphs.*

A set \mathcal{H} of copies of H in G is called uniformly colored (by the vertices of H) if for all copies $h, h' \in \mathcal{H}$ and vertices $a, a' \in V(H)$, having $h(a) = h'(a')$ implies $a = a'$.

Intuitively, if two copies overlap on some vertex v of G , they agree on the vertex a of H that is paired with v . Observe that the next lemma is a reformulation of a part of [9, Lemma 20] and is true on arbitrary graphs.

► **Lemma 4.2.** *Let \mathcal{H} be a set of N edge-disjoint copies of a graph H in a graph G .*

There exists a uniformly colored set $\mathcal{H}' \subseteq \mathcal{H}$ of cardinality at least $N/|H|^{|H|}$.

Proof. We want the inverse image of a given vertex of G under all copies of \mathcal{H}' to be either empty or a singleton. We consider random assignments of functions $f: V(G) \rightarrow V(H)$ and look at the probability that the copies of H in \mathcal{H} are *compatible with f* , meaning that for every $v \in V(H)$, $f(h(v)) = v$.

Write $\{h_1, \dots, h_N\}$ for \mathcal{H} and X_i for the random variable which is 1 if f is compatible with h_i and 0 otherwise. Since:

- the probability of $f(h_i(v)) = v$ for a given v is $1/|H|$,
- the probabilities of $f(h_i(v)) = v$ are pairwise independent for distinct values of v , and,
- having $X_i = 1$ is exactly having $f(h_i(v)) = v$ for all $v \in V(H)$,

the expected value of X_i is $1/|H|^{|H|}$. The number of copies compatible with f is given by $\sum_i X_i$, and its expected value is

$$\mathbb{E}[X] = \sum_i \mathbb{E}[X_i] = \sum_i \mathbb{P}[X_i = 1] = \frac{N}{|H|^{|H|}}.$$

Therefore, there must exist some f with at least this many uniformly colored copies of H . ◀

► **Definition 4.3.** Let \mathcal{H} be a set of copies of a graph H in a graph G .

A tree embedding \leq_T of $G[\mathcal{H}]$ is called uniformly-layered with regard to \mathcal{H} if \mathcal{H} is uniformly colored and for every color (with regard to \mathcal{H}), there is a level of \leq_T containing precisely the vertices of this color.

► **Lemma 4.4.** Let H and G be graphs, with G of treedepth d . Let \mathcal{H} be a uniformly-colored set of edge-disjoint copies of H in G . There exists $\mathcal{H}' \subseteq \mathcal{H}$ such that $|\mathcal{H}'| = \Theta_{|H|,d}(|\mathcal{H}|)$ with a uniformly-layered tree embedding \leq_T of $G[\mathcal{H}']$ with depth $|H|$.

Proof. Write k for $|H|$ and a_1, \dots, a_k for the vertices of H . Let G be a graph of treedepth d , and \mathcal{H} be a set of edge-disjoint and uniformly colored (by the vertices a_1, \dots, a_k) copies of H in G . Let \leq_T be a tree embedding of G with depth d . Let C_1, \dots, C_{d^k} be the set of all k -tuples of levels. For each copy h of H in \mathcal{H} , and for each $i \leq k$, there is a level ℓ_i such that $h(a_i)$ lies in level ℓ_i in \leq_T : to h we associate the tuple $C = (\ell_1, \dots, \ell_k)$.

Since there are d^k distinct tuples, there exists a k -tuple C associated to at least $|\mathcal{H}|/d^k$ copies. Let \mathcal{H}' be the subset of \mathcal{H} of copies associated to C . By construction, we have:

1. $|\mathcal{H}'| \geq |\mathcal{H}|/d^k$, and
2. \mathcal{H}' is uniformly layered² in $\leq_T^{\mathcal{H}'}$ (that is \leq_T restricted to the vertices of $G[\mathcal{H}']$). ◀

By combining this result with the lemmas from Section 3.1, testing H -freeness on graphs of treedepth at most d is reduced to testing H -freeness on graphs induced by many edge-disjoint copies of H , of treedepth at most $|H|$, with a uniformly layered treedepth embedding of depth $|H|$. We are more precise on the combined use of these lemmas in Section 4.3.

4.2 Constructing copies of H via parallel parts

► **Definition 4.5.** Let H be a graph with a total ordering π on its vertices.

A vertex v of H is called a source if it is not adjacent to any of its predecessors in π . Otherwise, it is referred to as an inner vertex.

Observe, in particular, that the minimal vertex in π is a source.

² An astute reader can notice that if $\ell_i = \ell_j$ for some pair i, j then one level can contain two or more colors. This is easily fixable by creating intermediate levels (e.g. 3.1, 3.2, ...) so that vertices of several colors in one level can choose arbitrarily their sub-level. E.g. if vertices blue and red are on level 3, we arbitrarily put the blue vertices on level 3.1, and the red one in 3.2. This keeps at most $|H|$ many (sub-)levels.

► **Definition 4.6.** Let H be a graph with a total ordering π on its vertices.

A parallel part of H is a maximal connected subgraph of H that contains no source. For convenience, the terms parallel part and part are used interchangeably.

► **Observation 4.7.** Any graph H with a total vertex ordering π has a unique vertex partition into sources and parallel parts. Moreover, the set of sources of H is an independent set in H .

Let H and G be graphs, with G induced by the edges of a set \mathcal{H} of edge-disjoint copies of H , and with a uniformly layered treedepth embedding of G of depth $|H|$.

We now consider the vertex ordering on H induced by the levels of the uniformly layered treedepth embedding of G . Observe that if $h \in \mathcal{H}$ and v is in a parallel part of H , then for every color, $h(v)$ has at most one neighbor of this specific color. Therefore, as the copies in \mathcal{H} are edge-disjoint, we get the following observation.

► **Observation 4.8.** Let P be a parallel part of H and h a copy of H in \mathcal{H} .

Every vertex in $h(P)$ is contained in exactly one copy of H in \mathcal{H} . In particular, if a vertex v of $G[\mathcal{H}]$ is contained in multiple copies from \mathcal{H} , then v is a (copy of a) source.

► **Definition 4.9.** Let F_1 and F_2 be two parallel parts of H and h_1 and h_2 be two copies of H in G . We say that $h_1(F_1)$ and $h_2(F_2)$ are compatible if for every source s adjacent to both F_1 and F_2 , we have $h_1(s) = h_2(s)$.

We now show how copies of H may be constructed from compatible parallel parts and their adjacent sources:

► **Observation 4.10.** Let F_1, \dots, F_n be the parts of H and S_1, \dots, S_n the sets of sources they are adjacent to, respectively. Given $h_1, \dots, h_n \in \mathcal{H}$ copies of H such that the graphs $h_i(F_i)$ are pairwise compatible, the graph $\cup_i h_i(F_i \cup S_i)$ is a copy of H in G (which may not be in \mathcal{H}).

We now introduce the notion of port. Informally, a port is a copy of a source from which a copy of a yet to be discovered parallel part of H may be found.

► **Definition 4.11.** Let $\mathcal{F} = \{F_i \mid i \in I\}$ be a set of distinct parts of H . A source s adjacent to $\cup_i F_i$ is called a port of \mathcal{F} if there exists a part $F \notin \mathcal{F}$ of H which is adjacent to s .

We show that, assuming H is connected and starting with the copy of a parallel part of H , we can always reconstruct a copy of H “blindly”, meaning that each time we consider the copy of a new part of H , we only choose the port from which we discover it.

Consider the following procedure:

■ **Algorithm 2** Constructing a copy of a connected component of H .

Input: A graph $G[\mathcal{H}]$ induced by a set \mathcal{H} of edge-disjoint copies.

A uniformly layered tree embedding \leq_T of depth $|H|$ of G .

A copy $h(F)$ of a parallel part of H contained in the component C_F of H .

Output: A set \mathcal{F} of copies of parts of H (including $h(F)$) forming a copy of C_F

- 1 $\mathcal{F} \leftarrow \{h(F)\};$
 - 2 **while** there exists a parallel part F' of C_F without a copy in \mathcal{F} **do**
 - 3 $h'(F') \leftarrow$ a copy of a parallel part of C_F without a copy in \mathcal{F} , and adjacent to a port of \mathcal{F} of maximal level in \leq_T ;
 - 4 $\mathcal{F} \leftarrow \mathcal{F} \cup \{h'(F')\};$
 - 5 **return** $\mathcal{F};$
-

► **Lemma 4.12.** *Algorithm 2 is sound and terminates.*

Proof. Termination is ensured by the fact that H , and by extension C_F , contain finitely many parallel parts. At the end of a run, \mathcal{F} contains precisely one copy of each part of C_F , including $h(F)$.

We prove soundness via the following invariant: at any step during a run of Algorithm 2, the set \mathcal{F} contains pairwise compatible copies of parts of C_F . We then conclude with Observation 4.10.

We proceed inductively over the while loop. Let \mathcal{F} be given after a partial execution of the algorithm. Since the algorithm is still running, there exists a part of C_F without a copy in \mathcal{F} . Let $h'(F') \in \mathcal{H}$ be a copy of a parallel part F' of C_F without a copy in \mathcal{F} , and such that F' is adjacent to a port s of \mathcal{F} of maximal level in \leq_T .

Assume for the sake of contradiction that $h'(F')$ is not compatible with some copy $h''(F'') \in \mathcal{F}$: let s' be a source of C_F adjacent with both F' and F'' but with $h'(s') \neq h''(s')$.

By construction of the algorithm, the graph G' induced by the copies of parts in $\mathcal{F} \cup \{h'(F')\}$ as well as by the corresponding copies of adjacent sources is connected.

Consider, in G' , a simple path P from $h'(s')$ to $h''(s')$ going through copies of parts of H via the ports that led to their addition to \mathcal{F} . Since initially $\mathcal{F} = \{h(F)\}$, such a path is well-defined.

Let s_m be the copy of a source of smallest level in \leq_T appearing on P . At s_m , P must change of parallel part. It quits a copy of a part F_1 to enter a copy of a part F_2 .

We perform case analysis depending on whether F_1 is discovered before F_2 by the algorithm or not.

Assume that F_2 is discovered after F_1 by the algorithm. Consider the sequence of ports that led to discoveries of parts traversed by P : $h'(s') = s_1, \dots, s_i, s_m, s'_1, \dots, s'_j = h''(s')$.

By construction of P , all sources among s_1, \dots, s_i are at a deeper level in \leq_T than s_m . Hence, all copies of parts of H that the path $h'(s') = s_1, \dots, s_i, s_m$ traverses, as well as $h'(F')$, should have been discovered before the copy of F_2 , a contradiction since $h'(F')$ is discovered after the copy of F_2 .

Assume that F_1 is discovered after F_2 by the algorithm. Similarly, as in the previous case, all copies of parts of H that the path $s_m, s'_1, \dots, s'_j = h''(s')$ traverses, as well as a copy of F' adjacent to $h''(s')$ should have been discovered before a copy of F_1 , a contradiction since no copy of F' adjacent to $h''(s')$ has been discovered. ◀

4.3 Testing H -freeness on graphs of bounded treedepth

To prove testability of H -freeness on graphs of bounded treedepth we proceed as follows. First, we prove that Algorithm 2 can be simulated inside a run of `RandomBoundedBFS`. Hence, the latter (with the right parameters) finds a copy of a given connected component of H with high probability. Then, by composing such calls sequentially as is done in `Tester`, we find H as a whole with high probability. Such a sequential composition is possible thanks to uniformly colored sets of copies. Indeed, the copies of connected components of H found by `RandomBoundedBFS` may use parallel parts of different copies, but are always compatible with the uniform coloring of copies of H , ensuring that found copies are disjoint. Finally, to ensure that we find H with probability $2/3$ and not $\Omega_{|H|, \epsilon}(1)$, we use the parameter n of `Tester` to strengthen probabilities.

Note that, the intermediate lemmas do not require G to have bounded treedepth, but $G[\mathcal{H}]$ to have a uniformly layered treedepth embedding. Thanks to Lemma 4.4 the former implies the latter.

► **Lemma 4.13.** *Let H and G be graphs, \mathcal{H} a set of edge-disjoint copies of a graph H , and \leq_T be a uniformly layered treedepth embedding of $G[\mathcal{H}]$ of depth $|H|$.*

Consider a run of RandomBoundedBFS (on $G[\mathcal{H}]$) with breadth $\Delta(H)$ and depth t . If, at step $n < t$ the algorithm has found the copy $h(u)$ of an inner vertex $u \in V(H)$ ($h \in \mathcal{H}$), then with probability $\Omega_H(1)$, the copy $h(N_H(u))$ of the neighborhood of u is discovered at step $n + 1$.

Proof. Since u is an inner vertex and \leq_T a uniformly layered treedepth embedding, the neighborhood of $h(u)$ in $G[\mathcal{H}]$ is precisely the image of $N_H(u)$ under h . Hence, in the step just after discovering u , RandomBoundedBFS finds all edges incident to u with probability at least $(\Delta_H - 1)! / \Delta_H^{\Delta_H - 1} = \Omega_H(1)$. ◀

► **Lemma 4.14.** *Let H and G be graphs, \mathcal{H} a set of edge-disjoint copies of a graph H , and \leq_T be a uniformly layered treedepth embedding of $G[\mathcal{H}]$ of depth $|H|$.*

Let s be a source of H , F a parallel part of H adjacent to s .

Consider a run of RandomBoundedBFS (on $G[\mathcal{H}]$) with breadth $\Delta(H)$ and depth t . If, at step $n < t$ the algorithm has found a copy $h(s)$ of s ($h \in \mathcal{H}$), then with probability $\Omega_H(1)$, there exists a copy $h' \in \mathcal{H}$ with $h'(s) = h(s)$ such that a copy of an inner vertex of F under h' is discovered at step $n + 1$.

Proof. Let F and F' be two distinct parallel parts, both adjacent to a source s of H , and let h be a copy of H in \mathcal{H} . Observe that there are as many copies of F as there are of F' which are adjacent to $h(s)$ in G (one each per copy with $h(s)$ in the image).

Let N be the number of copies of H $h(s)$ is in the image of. Let m_F be the number of edges of H in F which are incident to s . The probability that the next discovered edge does not connect to an inner vertex of some copy of F is at most $(N \cdot \deg^H(s) - N \cdot m_F) / (N \cdot \deg^H(s))$. So the probability that a discovered edge satisfies our requirements, when drawing Δ_H times, is at least $1 - \left(\frac{\deg^H(s) - m_F}{\deg^H(s)} \right)^{\Delta_H} = \Omega_H(1)$. ◀

► **Lemma 4.15.** *Let H and G be graphs, \mathcal{H} a set of edge-disjoint copies of a graph H , and \leq_T be a uniformly layered treedepth embedding of $G[\mathcal{H}]$ of depth $|H|$.*

Let F be a parallel part of H , and u an inner vertex of F .

Consider a run of RandomBoundedBFS (on $G[\mathcal{H}]$) with breadth $\Delta(H)$ and depth $t > |F|$. If, at step $n \leq t - (|F| - 1)$ the algorithm has already discovered the copy $h(u)$ of u ($h \in \mathcal{H}$), then with probability $\Omega_H(1)$, the copy $h(F)$ as well as the copies of the sources adjacent to F under h are discovered at step $n + |F| - 1$.

Proof. By applying Lemma 4.13 at most $|F|$ times, since $|F|$ bounds the diameter of F from above. ◀

► **Lemma 4.16.** *Let H and G be graphs, \mathcal{H} a set of edge-disjoint copies of a graph H , and \leq_T be a uniformly layered treedepth embedding of $G[\mathcal{H}]$ of depth $|H|$.*

Let u be an inner vertex of a connected component C of H .

Consider a run of RandomBoundedBFS (on $G[\mathcal{H}]$) with breadth $\Delta(H)$ and depth $t > |C|$. If the algorithm starts with a copy $h(u)$ of u ($h \in \mathcal{H}$), then with probability $\Omega_{\mathcal{H}}(1)$, the algorithm finds a copy of C consisting of copies of its parallel parts corresponding to elements of \mathcal{H} .

Proof. Lemmas 4.14 and 4.15 makes it possible to simulate the behavior of Algorithm 2 inside a run of the RandomBoundedBFS. With breadth $\Delta(H)$ and depth $|C|$ and probability $\Omega_H(1)$, the output of RandomBoundedBFS contains a copy of C in G which is constructed just as in the algorithm. Transitions between parts is ensured by Lemma 4.14. ◀

► **Theorem 4.17.** *For any graph H and graph class \mathcal{C} of bounded treedepth, the property of being H -free can be tested in the random neighbor model on graphs of \mathcal{C} with constant query complexity and one-sided error.*

Proof. Fix a graph $G \in \mathcal{C}$ which is ϵ -far from being H -free. We prove that there exists a constant $n = O_{\epsilon, H, \mathcal{C}}(1)$ such that `Tester`(n, H, ϵ, G) rejects with probability at least $2/3$. Since `Tester` starts by stripping isolated vertices from H , we assume that H contains none. Let n_H be the number of connected components of H . First, we use Lemmas 3.1, 3.4, 4.2, and 4.4 (in this order) to obtain a set \mathcal{H} of cardinality $\Omega_{\epsilon, H, \mathcal{C}}(|V(G)|)$ such that $G[\mathcal{H}]$ is c -degree preserving and contains a uniformly-layered tree embedding³ (where $c = O_{\epsilon, H, \mathcal{C}}(1)$).

Let C be a connected component of H . We prove that the output of a single call to `RandomBoundedBFS`($G[\mathcal{H}], |H|, \Delta(H)$) contains a copy of C compatible with the uniformly-layered tree embedding with probability $\Omega_{\epsilon, H}(1)$. By Lemma 4.16, it is sufficient to prove that `RandomBoundedBFS` starts on the copy of an inner vertex of C with high probability. Since C contains an edge, it cannot be reduced to a single source, and C contains at least one inner vertex. As copies of inner vertices are in the image of precisely one copy of H , and as $G[\mathcal{H}]$ contains $\Omega_{\epsilon, H, \mathcal{C}}(|G|)$ copies of H , $G[\mathcal{H}]$ contains $\Omega_{\epsilon, H, \mathcal{C}}(|G|)$ vertices copies of inner vertices of C . Hence, `RandomBoundedBFS`($G[\mathcal{H}], |H|, \Delta(H)$) starts on an inner vertex of C with probability $\Omega_{\epsilon, H, \mathcal{C}}(1)$.

Now, with Lemma 3.3, and the fact that \mathcal{H} is c -degree preserving, we have that the output of a single call to `RandomBoundedBFS`($G, |H|, \Delta(H)$) contains a copy of C compatible with the uniformly-layered tree embedding with probability $\Omega_{\epsilon, H, \mathcal{C}}(1)$ as well.

Each execution of the part of the algorithm which is repeated n times (i.e. the n_H calls to `RandomBoundedBFS`) finds a copy of H with probability $p = \Omega_{\epsilon, H, \mathcal{C}}(1)$ each time it is executed. Indeed, the i th connected component is found with probability $\Omega_{\epsilon, H, \mathcal{C}}(1)$ by the i th call to `RandomBoundedBFS`, in the form of a copy compatible with the uniformly-layered tree embedding, so these copies of the connected components of H found with high probability are vertex disjoint and form a copy of H . Let n be such that $2/n \leq p$. Then, the probability that `Tester`(n, H, ϵ, G) finds a copy of H is at least $1 - (1 - 2/n)^n > 1 - e^{-2} > 2/3$. ◀

5 Outlook

Could one extend our results to broader classes of graphs? At least not in the direction of bounded degeneracy. In [7] it is proved that for any $r \geq 4$, C_r -freeness is not testable for graphs of bounded $(\lfloor r/2 \rfloor - 1)$ -admissibility. This being a special case of bounded degeneracy.

However, bounded r -admissibility is incomparable with nowhere denseness. Nowhere dense is a very robust notion that yield numerous algorithms in recent years. A key difference with bounded expansion, is that while nowhere dense graphs enable p -treedepth coloring, it requires $O_{\delta, p}(|G|^\delta)$ many colors (and not constantly many). Meaning that constant numbers in some of our key lemmas would now depend on G . Additionally, nowhere dense graphs can have a super-linear number of edges (e.g. $|G| \cdot \log(|G|)$). One could start by adapting the concept of ϵ -far, enabling removal of an edge set with size $O_\delta(\epsilon \cdot |G|^{1+\delta})$ for all $\delta > 0$.

Another extension, as in [10], could be to look at approximate proof labelling scheme. However, the main result of [10] (pushing from finitely many forbidden subgraphs to monotone properties) cannot be adapted to bounded expansion. In part because already bipartiteness is not testable on bounded degree graphs, so even less so on graphs with bounded expansion.

³ Note that having uniformly-layered tree embedding is a monotone property, so restricting further to get c -degree preserving is safe.

References

- 1 Isolde Adler, Noleen Köhler, and Pan Peng. On testability of first-order properties in bounded-degree graphs and connections to proximity-oblivious testing. *SIAM J. Comput.*, 53(4):825–883, 2024. doi:10.1137/23M1556253.
- 2 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It’s all about regularity. *SIAM J. Comput.*, 39(1):143–167, 2009. doi:10.1137/060667177.
- 3 Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM J. Comput.*, 37(6):1703–1727, 2008. doi:10.1137/06064888X.
- 4 Noga Alon and Asaf Shapira. Every monotone graph property is testable. *SIAM J. Comput.*, 38(2):505–522, 2008. doi:10.1137/050633445.
- 5 Christine Awofeso, Patrick Greaves, Oded Lachish, Amit Levi, and Felix Reidl. A sufficient condition for characterizing the one-sided testable properties of families of graphs in the random neighbour oracle model, 2025. doi:10.48550/arXiv.2511.19027.
- 6 Christine Awofeso, Patrick Greaves, Oded Lachish, Amit Levi, and Felix Reidl. Testing Ck-freeness in bounded admissibility graphs. In *Intl. Coll. on Automata, Languages and Programming (ICALP)*, volume 334 of *LIPICs*, pages 15:1–15:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.ICALP.2025.15.
- 7 Christine Awofeso, Patrick Greaves, Oded Lachish, and Felix Reidl. Results on h-freeness testing in graphs of bounded r-admissibility. In *Symp. on Theoretical Aspects in Computer Science (STACS)*, volume 327 of *LIPICs*, pages 12:1–12:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.STACS.2025.12.
- 8 Artur Czumaj, Morteza Monemizadeh, Krzysztof Onak, and Christian Sohler. Planar graphs: Random walks and bipartiteness testing. *Random Struct. Algorithms*, 55(1):104–124, 2019. doi:10.1002/RSA.20826.
- 9 Artur Czumaj and Christian Sohler. A characterization of graph properties testable for general planar graphs with one-sided error (it’s all about forbidden subgraphs). In *Symp. on Foundations of Computer Science (FOCS)*, 2019. doi:10.1109/FOCS.2019.00089.
- 10 Louis Esperet and Sergey Norin. Testability and local certification of monotone properties in minor-closed classes. In *Intl. Coll. on Automata, Languages and Programming (ICALP)*, volume 229 of *LIPICs*, pages 58:1–58:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.ICALP.2022.58.
- 11 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.
- 12 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.
- 13 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. definition bounded degree model. doi:10.1007/S00453-001-0078-7.
- 14 Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion I. Decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008. doi:10.1016/J.EJC.2006.07.013.
- 15 Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM J. Comput.*, 42(3):1095–1112, 2013. doi:10.1137/120890946.
- 16 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.