

Tolerant Testers for Subgraph-Freeness

Reut Levi 

Efi Arazi School of Computer Science, Reichman University, Herzliya, Israel

Jonathan Meiri 

Efi Arazi School of Computer Science, Reichman University, Herzliya, Israel

Abstract

In this paper we study the problem of tolerantly testing the property of being H -free (which also implies distance approximation from being H -free).

In the general-graphs model, we show that for tolerant K_k -freeness testing can be achieved with query complexity that is polynomial in the arboricity of the input graph G , $arb(G)$, and independent of the size of G (for graphs in which the average degree is $\Omega(1)$).

Specifically for triangles, our algorithm distinguished graphs which are ϵ -close to being triangle-free from graphs that $3\epsilon(1 + \eta)$ -far from being triangle-free with expected query complexity which is $\tilde{O}(arb^3(G))$ (for constant η and ϵ).

For general k -cliques our algorithm distinguishes graphs which are ϵ -close to being K_k -free from graphs which are $\binom{k}{2}\epsilon(1 + \eta)$ -far from being K_k -free with expected query complexity which is polynomial in k , ϵ , γ and $arb(G)$.

We then generalize our result and provide a similar result for any motif H which is 2-connected of radius 1. This includes for example the wheel-graph.

Finally, we show that our tester can be applied to the bounded-degree model for tolerantly testing H -freeness for any motif H . The query complexity of the algorithm is polynomial in the degree bound, d , improving the previous state-of-the-art by Marko and Ron (TALG 2009) that obtained quasi-polynomial query complexity in d .

2012 ACM Subject Classification Theory of computation \rightarrow Streaming, sublinear and near linear time algorithms

Keywords and phrases Tolerant Testing, Property Testing, Subgraph freeness, distance approximation, arboricity

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.77

Funding *Reut Levi*: The author was supported by the Israel Science Foundation under Grant 1867/20.

Jonathan Meiri: The author was supported by the Israel Science Foundation under Grant 1867/20.

1 Introduction

The problem of testing H -freeness, where H is a small *motif* (namely, of size which is independent of the input graph), has received a lot of attention due to its fundamental importance for various fields including biology, sociology and network science. Detecting specific patterns, such as subgraphs, can reveal community structures, vulnerabilities, or anomalous behavior.

In the realm of property testing the decision problem is relaxed such that we are only aiming to distinguish graphs which are H -free from graphs that are *far* from being H -free (according to some predetermined distance measure). However, handling real-world data often means dealing with noise, incompleteness, or small errors. In these scenarios, tolerant testing (introduced by Parnas, Ron and Rubinfeld [15]) for H -freeness becomes crucial, as it allows us to distinguish graphs that are nearly H -free from those that are far from being H -free, thereby bypassing the problem of imperfections in the data. More specifically, a tolerant property testing algorithm is required to distinguish objects that are ϵ_1 -close to having a given property \mathcal{P} from objects that are ϵ_2 -far from having \mathcal{P} , for some parameters



© Reut Levi and Jonathan Meiri;
licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 77; pp. 77:1–77:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$0 \leq \epsilon_1 < \epsilon_2 \leq 1$. Clearly, by definition, tolerant-testing is at least as hard as testing. In fact, the separation between tolerant and non-tolerant testing can be very dramatic [8]. Another benefit of tolerant testing is that it implies (by a simple reduction) distance approximation where the ratio of approximation depends on the parameters ϵ_1, ϵ_2 (see more details in [15]). Therefore, whenever feasible, we should aim to develop tolerant testers, ideally without incurring a significant overhead in query complexity.

One of the early works on tolerant testing by Marko and Ron [13] showed that it is possible to tolerantly test H -freeness in the bounded-degree model [9] for any motif H with query complexity that is quasi-polynomial in the degree bound and independent of the size of the graph.

In the regime of non-tolerant testers, Levi [12] showed that it is possible to test triangle-freeness with query complexity that depends linearly in the arboricity of the graph (which is tight for graphs with arboricity $O(\sqrt{n})$) and is independent of the size of the graph (assuming the average degree is $\Omega(1)$). Since graphs of bounded degree have bounded arboricity, this result applies to a broader family of graphs. However, this tester is not tolerant.

One natural question is whether the result in [12] can be extended to show that tolerant testing of triangle-freeness (and ideally other motifs) is possible in the general-graphs model with query complexity that depends only on the arboricity of the graph. Such a result would also generalize [13] (since bounded-degree graphs inherently have bounded arboricity). Another important question is whether the query complexity of subgraph freeness in the bounded-degree model can be improved to be polynomial in the degree bound (rather than quasi-polynomial). In this paper, we answer both questions affirmatively.

1.1 Our Results

1.1.1 The general-graphs model

1.1.1.1 Tolerant Testing K_k -freeness

Our tester for the property of being K_k -free distinguishes graphs that are ϵ -close to being K_k free from graphs that are $\epsilon \binom{k}{2} (1 + \eta)$ -far from being K_k -free, where $\epsilon \in (0, 1]$ and $\eta \in (0, 1]$ are parameters. The expected query complexity of the algorithm is $\tilde{O}\left(\frac{k^4 \theta^{2k-3}}{\eta^2 \epsilon \bar{d}} + k^5 \theta^{2k-4} \theta^{\min\{2, k-2\}}\right)$ where $\theta = \Theta(\text{arb}(G)/(\eta\epsilon))$ and \bar{d} is the average degree (see Theorem 21).

Specifically, assuming $\bar{d} = \Omega(1)$, for the case of triangles, our algorithm distinguished graphs which are ϵ -close to being triangle-free from graphs that $3\epsilon(1 + \eta)$ -far from being triangle-free with expected query complexity which is $\tilde{O}(\text{arb}^3(G)) \cdot \text{poly}(\epsilon^{-1}, \eta^{-1}, k)$.

For k -cliques where $k > 3$, our algorithm distinguishes graphs which are ϵ -close to being K_k -free from graphs which are $\binom{k}{2}\epsilon(1 + \eta)$ -far from being K_k -free with expected query complexity which is $\tilde{O}(\text{arb}^{2k-2}(G)) \cdot \text{poly}(\epsilon^{-1}, \eta^{-1}, k)$.

1.1.1.2 Tolerant H -freeness for H which is 2-connected with radius 1

We extend our result and apply our tester for motifs that are 2-connected with radius 1. This includes for example the wheel-graph¹. Specifically our tester distinguishes graphs that are ϵ -close to being H free from graphs that are $\epsilon|E(H)|(1 + \eta)$ -far from being H -free. The expected query complexity of the algorithm is $\tilde{O}\left(h^7/\eta^2 + h^3/(\eta\epsilon\bar{d})\right) \cdot \theta^{3h-2}$ where $\theta = \Theta(\text{arb}(G)/(\eta\epsilon))$ and $h = |V(H)|$ (see Theorem 28).

¹ Any connected graph with an additional vertex that is incident to all other vertices is 2-connected with radius 1.

Specifically, assuming $\bar{d} = \Omega(1)$, the expected query complexity of our tester is $\tilde{O}(\text{arb}^{3h-2}(G)) \cdot \text{poly}(\epsilon^{-1}, \eta^{-1}, h)$.

1.1.2 The bounded-degree graphs model

Tolerant H -freeness for any H

We show that our tolerant tester can be applied to the bounded-degree model to test H -freeness for any motif H . Specifically the tester distinguishes graphs that are ϵ -close to being H free from graphs that are $\epsilon|E(H)|(1 + \eta)$ -far from being H -free. The expected query complexity of the algorithm is $\tilde{O}((h^7/\eta^2 + h^3/(\eta\epsilon\bar{d})) \cdot d^{3h-2})$ where $h = |V(H)|$ and d is the degree bound.

Specifically, assuming $\bar{d} = \Omega(1)$, the expected query complexity of our tester is $\tilde{O}(d^{3h-2}) \cdot \text{poly}(\epsilon^{-1}, \eta^{-1}, h)$.

This improves the result in [13] that has query complexity that is quasi-polynomial in d to polynomial dependence in d .

► **Remark 1.** As noted in [13], the result of Bogdanov, Obata and Trevisan [2] implies a linear lower bound on the query complexity of approximating the distance from being triangle-freeness for some small multiplicative and/or additive error (even in bounded degree graphs). Therefore, we can not hope to obtain tolerant testers of subgraph freeness for all parameters ϵ_1, ϵ_2 .

► **Remark 2.** As shown in [3], testing C_4 -freeness requires $\Omega(n^{1/4})$ queries. Therefore we can not hope to obtain tolerant testers (in the general-graphs model) for general motifs whose query complexity depends only on the arboricity of the graph.

1.2 Our Algorithm

In this section we describe our algorithm for tolerantly test K_k -freeness. This gives a good perspective of our approach. The first ingredient of our algorithm, which also appears in [12], is to perform the test on a subgraph of G . This subgraph, which we refer to as G_θ is the graph after removing the edges between θ -heavy vertices, where a vertex is θ -heavy if its degree is at least θ where $\theta = \Theta(\text{arb}(G)/(\eta\epsilon))$. By doing this modification we might decrease the distance by at most $\Theta(\eta\epsilon)$, so if the graph is sufficiently far from being K_k -free, we will still have enough witnesses of violation. On the positive side, since this subgraph does not have edges in which both endpoints are θ -heavy, any K_k in this graph will include at most a single θ -heavy vertex and therefore can be revealed by exploring the neighborhood of the non-heavy vertices of the clique.

The second ingredient of our algorithm is to approximate the number of copies of K_k in G_θ (it will become clear later why we need to approximate this parameter). This can be done by sampling vertices, v , u.a.r. from G , checking if the degree of v is bounded by θ and if so selecting $k - 1$ random indices in $[\theta]$. We then perform $k - 1$ neighbor-queries on v , according to the selected indices, and check if the subgraph induced on these neighbors (if such neighbors exist) induces a $k - 1$ -clique. Thus, our sample space is of size $\Theta(n\theta^{k-1})$ and if the graph is ϵ -far from being K_k -free then we have at least ϵm copies of K_k so we need $\Theta(n\theta^{k-1}/(\epsilon m))$ (which is $O(\text{arb}(G)^{k-1}/\epsilon)$ when $\bar{d} = \Omega(1)$) trials to get a good approximation.

Before we describe the last ingredient of our algorithm we shall define the H -copies graph of G_θ , which we denote by $(G_\theta)^H$. In this graph the vertex set is the set of H -copies in G_θ and a pair of copies are adjacent iff they share at least one edge. Consider a maximal independent set (MIS), S , of this graph. By definition, the copies of this set are edge-disjoint.

Thus the minimum number of edges we need to remove from G_θ in order to make it H -free, which we denote by $\text{dist}_H(G_\theta)$, is at least $|S|$. On the other hand, by the maximality of S it follows that $\text{dist}_H(G_\theta) \leq |E(H)||S|$ (since we can remove all H -copies in G_θ by removing all the edges of the copies in S). Therefore if we can get a good approximation of *any* MIS of $(G_\theta)^H$, then we will get a good approximation (up to a $|E(H)|$ -factor) to the distance of G_θ from being H -free. This is also true for any MIS that is constructed greedily according to some order on the vertices.

Fortunately, as shown by the ingenious work of Yoshida, Yamamoto and Ito [17], if we pick a random vertex v in a graph and a random ordering on the vertices of the graph then the expected number of recursive calls we need to perform in order to locally simulate the outcome of the greedy MIS algorithm on that graph (according to the random ordering we picked) equals the average degree of the graph plus 1. Therefore, we can get a good approximation to the *average* probability that a random H -copy belongs to a random greedy MIS (which is determined by a random ordering) by executing the local simulation on random H -copies and a random orderings (*both* selected uniformly and independently on each trial). Since the ratio between any MIS in a graph F to the total number of vertices in F is at most $1/(\Delta(F) + 1)$, where $\Delta(F)$ denotes the maximum degree in F , we obtain that it suffices to evaluate the outcome of a greedy MIS (each time according to a new random ordering) on $\Theta(\Delta((G_\theta)^H))$ copies selected uniformly at random. This will give us a good approximation to the *average* probability that a random H -copy belongs to a random greedy MIS. When we multiply this by the approximation to the total number of H -copies we previously obtained, this will provide a good approximation to the distance of G from being H -free.

1.2.1 The extension for other motifs

We describe the above mentioned algorithm as a general framework for tolerant testing of H -freeness and then implement (according to the defined interface) the necessary specific procedures for testing K_k -freeness and then provide more general procedures for H -freeness of any motif H which is 2-connected and has radius 1. Finally, we show that the procedures for the latter family of motifs can be used to test H -freeness for any motif H in the bounded degree model.

1.3 Related Work

1.3.1 Tolerant testing of H -freeness in the bounded degree model

As mentioned-above Marko and Ron [13] studied the problem of tolerantly testing H -freeness in the bounded degree model. They begin by describing a global algorithm that constructs a cover of edges by first performing $O(\log d)$ iterations in which the edges of some of the H -copies are added to the cover by a random process (that can be viewed as a distributed randomized algorithm for constructing a large independent set of copies) and then a single edge is added to the cover from each one of the uncovered copies. Since the global algorithm has a distributed nature they show they can locally simulate its outcome by performing $d^{O(\log d)}$ queries and therefore obtain a good approximation to the size of the cover which gives a good approximation to the distance from being H -free.

1.3.2 Non-tolerant testing of subgraph-freeness

In the bounded degree model Goldreich and Ron [9] observed that it is possible to test triangle freeness with query complexity $O(1/\epsilon)$ in graphs of maximum degree bounded by some constant.

The problem of testing triangle freeness in the general graph model was first studied by Alon, Kaufman, Krivelevich, Ron [1]. The query complexity of their algorithms depends on n and \bar{d} , the number of vertices in the graph and the average degree, respectively. They provided sublinear upper bounds for almost the entire range of parameters. Moreover, their upper bounds are at most quadratic in their lower bounds. However, they are tight only when $d_{\max} = O(\bar{d})$ and $\bar{d} \leq \sqrt{n}$, where d_{\max} denotes the maximum degree and \bar{d} denotes the average degree of the graph or when $\bar{d} = \Theta(1)$. Shortly after, Rast [16] and Gugelmann [10] improved the upper bound and lower bound of [1], receptively, for some ranges of the parameters.

More recently, Levi [12] studied the complexity of testing triangle freeness as a function of the arboricity of the graph (in the general graphs model). It was shown that the complexity of testing triangle freeness depends linearly in the arboricity of the graph for graphs in which the arboricity is $O(\sqrt{n})$ and the average degree is $\Omega(1)$.

Even more recently, Eden, Levi and Ron [3] studied the problem of testing C_k -freeness and showed that if the motif is not a k -clique then it is not always possible to provide a tester whose query complexity is independent of the size of the graph. In particular, they showed that testing C_4 -freeness requires $\Omega(n^{1/4})$ queries.

1.3.3 Sublinear algorithms that receive the arboricity of the graph as a parameter

Several sublinear-time graph algorithms for counting and sampling give improved results when the graph G has bounded arboricity. This includes the following results. Eden, Ron and Rosenbaum [4] designed an algorithm for sampling edges almost uniformly.

Eden, Ron and Seshadhri [6] estimate the degree distribution moments of an undirected graph and in particular estimate the average degree of a graph.

In another paper, Eden, Ron and Seshadhri [7] give a $(1 \pm \epsilon)$ -approximation for the number of k -cliques. In a more recent work, Eden, Ron, and Rosenbaum [5] showed that in this setting sampling cliques is harder than counting.

2 Preliminaries

We consider the general-graphs model [14, 11], where the algorithm can perform the following types of queries:

1. For any $v \in V$, query for v 's degree;
2. For any vertex $v \in V$ and index i , query for v 's i -th neighbor (if $i > d(v)$, then a special symbol is returned);
3. For any pair of vertices u, v , query whether $\{u, v\} \in E$.

We denote by $\Delta(G)$ the maximum degree of the graph G . We denote by $\delta_H(G)$ the distance of G from being H -free, namely, the fraction of edges that we need to remove from G to make it H -free. We denote by $\text{dist}_H(G) = \delta_H(G)|E(G)|$, namely, the minimum number of edges we need to remove from G in order to make it H -free. For a graph F we use $V(F)$ to denote its vertex set and $E(F)$ to denote its edge set.

When we say with high constant probability (w.h.c.p.) we mean we can adjust the high constant probability without changing the asymptotic complexity of the algorithm.

► **Definition 3** (Copies of a motif H). *For a graph G and a motif H , we say that a subgraph \mathcal{H} of G is a copy of H in G if \mathcal{H} is isomorphic to H .*

We use $n_H(G)$ to denote the number of H -copies in G .

Given a threshold θ , we call the vertices, v , in G such that $d_G(v) > \theta$, θ -heavy and otherwise we call them θ -light.

► **Definition 4** (The graph G_θ). For a graph $G = (V, E)$ and a threshold θ , the graph G_θ is the graph whose vertex set is V and its edge set is:

$$E \setminus \{\{u, v\} : u \text{ and } v \text{ are } \theta\text{-heavy}\}.$$

► **Definition 5.** The arboricity of a graph G , denoted by $\text{arb}(G)$, is the minimum number of forests into which the edges of the graph can be partitioned.

It is known that for $\theta = \text{arb}(G)/\eta$ where $\eta \in (0, 1/2]$, $|E(G_\theta)| \geq (1 - 2\eta)m$ (see, e.g., Claim 4 in [12]).

We assume that the arboricity of the input graph G as well as the number of edges, m , and vertices, n , are known to the algorithm.

► **Remark 6.** If the algorithm is not provided with the arboricity of the graph, then it may run the procedure from [12] that obtains the *effective arboricity* of the graph. More specifically, it was shown in [12] how to obtain a value α^* that with high constant probability satisfies the following: (1) $\alpha^* \leq 2\text{arb}(G)$; (2) The number of edges between vertices whose degree is at least $\Theta(\alpha^*/\epsilon)$ is at most ϵm , which suffices for our needs. Up to polylogarithmic factors in n and polynomial factors in $1/\epsilon$, the query complexity of the procedure is $O(\text{arb}(G))$ assuming the average degree is $\Omega(1)$.

2.1 The H -copies graph of G

In order to describe our algorithm we shall use the following definitions.

► **Definition 7.** Given a graph $G = (V, E)$ and a motif H we define the H -copies graph of G , G^H to be the graph whose vertex set are the copies of H in G , $\mathcal{H}_1, \dots, \mathcal{H}_\ell$ and $e = \{\mathcal{H}_1, \mathcal{H}_2\}$ belongs to the edge set of G^H iff \mathcal{H}_1 and \mathcal{H}_2 share at least one edge.

Next, we define the notion of an estimator for the number of H -copies of a graph G . The estimator is given a guessed lower bound on $n_H(G_\theta)$. If the guess is accurate, the estimator produces a reliable estimate; otherwise, it does not significantly overshoot, as detailed in the following definition.

► **Definition 8.** We say that an algorithm is a H -copies number estimator for a graph G if it receives as parameters s , a guess of a lower bound on $n_H(G)$, and γ , the approximation parameter, and possibly other parameters for which the following holds: If $s \leq n_H(G)$ then w.h.c.p. the return value of the algorithm is in $(1 \pm \gamma)n_H(G)$. Otherwise, w.h.c.p. it is at most $(1 + \gamma)s$.

► **Remark 9.** The role of the guess is to provide some (indirect) control to the user of the estimator on the number of attempts to sample a copy from the graph by the estimator (e.g. if the graph is H -free then the estimator might make too many attempts to sample a H -copy with no success).

► **Definition 10.** An algorithm \mathcal{A} is a H -copies oracle of a graph $G = (V, E)$ if it supports the following queries.

1. On query Random-copy with parameters $t \in \mathbb{N}$ and $\delta \in (0, 1]$, it returns a copy of H in G uniformly at random or fails. If the number of copies in G is at least t then it returns a copy w.p. at least $1 - \delta$.
2. On query All-neighbors(\mathcal{H}), where \mathcal{H} is a copy of H in G , it returns all the neighbors of \mathcal{H} in G^H (see Definition 7).

2.2 Local Simulation of Greedy Maximal IS

Our algorithm uses a local simulation of a greedy MIS algorithm. Algorithm 1 describes this local simulation. For a vertex set V , we let S_V denote the set of all permutations over V .

■ **Algorithm 1** Local-Simulation-Greedy-MIS.

Input: Query access to a graph $G = (V, E)$ and parameters $\pi \in S_V$ and $v \in V$

1. Perform an All-Neighbors query on v and sort its neighbors according to π . Let v_1, \dots, v_ℓ denote its neighbors according to this order.
 2. For $i = 1, \dots, \ell$:
 - a. If v_i precedes v in π , recursively call Local-Simulation-Greedy-MIS with parameters π and v_i . If the return value is *YES* then return *NO*.
 3. Return *YES*.
-

Let $R_\pi^G(v)$ denote the number of (recursive) calls to Local-Simulation-Greedy-MIS during the evaluation of Local-Simulation-Greedy-MIS on G , π and v . We shall use the following result from the work of Yoshida, Yamamoto and Ito [17].

► **Theorem 11** ([17]). *For any graph $G = (V, E)$ with n vertices and m edges,*

$$\mathbb{E}_{\pi \in S_V, v \in V} [R_\pi^G(v)] \leq 1 + \frac{m}{n}.$$

3 The algorithm for tolerant testing of H -freeness

Before describing our algorithm we first relate the distance from being H -free to the size of a Maximal-Independent-Set of G^H as stated in the following claim. We then use this relation to provide a relation between the expectation of a random variable to the distance from being H -free.

▷ **Claim 12.** For any maximal independent set of G^H , S , it holds that

$$|S| \leq \text{dist}_H(G) \leq |S| \cdot |E(H)|.$$

Proof. Since S is an independent set in G^H , it follows that any pair $\mathcal{H}_1, \mathcal{H}_\epsilon \in S$ is a pair of edge-disjoint copies of H in G (by definition). Since S is maximal, it follows that if we remove for each $\mathcal{H} \in S$ all its edges from G then G becomes H -free (since we remove at least one edge from each copy of H in G). ◁

▷ **Claim 13.** For v that is drawn u.a.r. from $V(G^H)$ and π which is drawn u.a.r. from S_V it follows that:

$$\Pr[v \in \text{MIS}_\pi(G^H)] \in \left[\frac{\text{dist}_H(G)}{|E(H)|n_H(G)}, \frac{\text{dist}_H(G)}{n_H(G)} \right]. \quad (1)$$

Moreover,

$$\Pr[v \in \text{MIS}_\pi(G^H)] \geq 1/(\Delta(G^H) + 1). \quad (2)$$

Proof. By Claim 12, for any maximal IS in G^H , S , it holds that $\frac{\text{dist}_H(G)}{|E(H)|} \leq |S| \leq \text{dist}_H(G)$. Therefore, for any such S it holds that $\Pr[v \in S] \in \left[\frac{\text{dist}_H(G)}{|E(H)|n_H}, \frac{\text{dist}_H(G)}{n_H} \right]$, where v is drawn u.a.r. from $V(G_H)$. This is true since $|V(G_H)| = n_H$. Since $\text{MIS}_\pi(G^H)$ is maximal for any

π (by definition), the claim follows. Equation 2 follows from the fact that each vertex v in the independent set covers at most $d(v) + 1$ vertices from the graph. Thus, any maximal independent set is of size at least $n_H(G)/(\Delta(G^H) + 1)$, in particular, $MIS_\pi(G^H)$ (for any π). This concludes the proof. \triangleleft

By using the previous claims we arrive to the following claim.

\triangleright **Claim 14.** If $s \geq n_H(G)$ then with high constant probability, the return value of Approximate-Average-MIS-In-Auxiliary-Graph is in

$$\left[(1 - \gamma) \frac{\text{dist}_H(G)}{|E(H)|n_H(G)}, (1 + \gamma) \frac{\text{dist}_H(G)}{n_H(G)} \right]. \quad (3)$$

Proof. Let E_1 denote the event that in all the executions of Step 2a of the algorithm, a copy of H was returned. By definition 10, the setting of δ and since $s \geq n_H(G)$, w.h.c.p., E_1 occurs. We henceforth assume E_1 occurs. Consider the random variable Z_i (see Step 2c of the algorithm). By construction Z_i takes values in $\{0, 1\}$. By Claim 13, Equation 1, $\mathbb{E}(Z_i) \in \left[\frac{\text{dist}_H(G)}{|E(H)|n_H(G)}, \frac{\text{dist}_H(G)}{n_H(G)} \right]$. By Equation 2, $\mathbb{E}(Z_i) \geq 1/(\Delta(G^H) + 1)$. The claim follows from the setting of ℓ and the multiplicative Chernoff's bound (see Theorem 30). \triangleleft

■ **Algorithm 2** Approximate-Average-MIS-In-Auxiliary-Graph.

Input: γ - approximation parameter, s - guess on a lower bound on $n_H(G)$, Δ - an upper bound on $\Delta(G^H)$, and access to H -copy oracle and H -copies estimator of G (see Definitions 8, 10)

1. Set $\ell = \Theta(\Delta/\gamma^2)$ and $\delta = \Theta(1/\ell)$.
2. For $i = 1, \dots, \ell$
 - a. Run a H -copy oracle of G with parameters s and δ , and obtain a H -copy, \mathcal{H} , uniformly at random.
 - b. Draw π u.a.r. and run Algorithm 1 on \mathcal{H} , π and the H -copies oracle of G , (see Definition 7).
 - c. Set $Z_i = 1$ if the algorithm returned YES and set $Z_i = 0$ otherwise.
3. Return $\hat{Z} = \sum_{i \in [\ell]} Z_i / \ell$.

■ **Algorithm 3** Tolerant-Test-Subgraph-freeness.

Input: ϵ, η and query access to a graph G

1. Set $\gamma = \eta/64$ and $\theta = \text{arb}(G)/(2\gamma\epsilon)$
2. Run the number of H -copies estimator for the graph G_θ with parameters $\epsilon m/2$ and γ (see Definition 8) and obtain \hat{n} .
3. If \hat{n} is less than ϵm then return ACCEPT.
4. Run Approximate-Average-MIS-In-Auxiliary-Graph on the graph G_θ with parameters $\epsilon m/2$ and γ and $\Delta((G_\theta)^H)$. Let \hat{Z} denote the returned value.
5. if $\hat{n}\hat{Z} > \epsilon m(1 + \gamma)^2$ then return REJECT.

We next prove our main theorem for this section. In order to realize this theorem we provide in the next sections concrete H -copies oracles and H -copies number estimators (first for H which is a k -clique and then generalize, for H which is 2-connected and has radius 1).

► **Theorem 15.** *Given parameters $\epsilon \in (0, 1]$, $\eta \in (0, 1]$ and query access to a graph G , Tolerant-Test-Subgraph-freeness has the following guarantees. If G is ϵ -close to being H -free then w.h.c.p. it accepts G . If G is $\epsilon|E(H)|(1 + \eta)$ -far from being H -free then w.h.c.p. it rejects G .*

Proof. Assume G is ϵ -close to being H -free. If the algorithm accepts G in Step 3 then we are done. Otherwise, it follows that $\hat{n} \geq \epsilon m$.

Let E_1 denote the event that the estimator in Step 2 returns a good approximation as described in Definition 8. Conditioned on E_1 and the fact that $\hat{n} \geq \epsilon m$ it is implied that $n_H(G_\theta) \geq \epsilon m/2$ (assume otherwise and reach a contradiction to the fact that $\hat{n} \geq \epsilon m$ as $(1 + \gamma) < 2$) and hence its return value is in $(1 \pm \gamma)n_H(G_\theta)$.

Let E_2 denote the event that Approximate-Average-MIS-In-Auxiliary-Graph returns a good approximation as described in Claim 14. Conditioned on $E_1 \cap E_2$ we obtain that

$$\hat{n}\hat{Z} \in \left[(1 - \gamma)^2 \frac{\text{dist}_H(G_\theta)}{|E(H)|}, (1 + \gamma)^2 \text{dist}_H(G_\theta) \right]. \quad (4)$$

Since $\text{dist}_H(G_\theta) \leq \text{dist}_H(G) \leq \epsilon m$ we obtain that $\hat{n}\hat{Z} \leq (1 + \gamma)^2 \epsilon m$ and hence the algorithm accepts G .

Assume G is β -far from being H -free where $\beta = \epsilon|E(H)|(1 + \eta)$. By the setting of θ it follows that $\text{dist}_H(G_\theta) \geq (1 - \gamma)\beta m$. Thus, $n_H(G_\theta) \geq (1 - \gamma)\beta m$. Therefore, conditioned on $E_1 \cap E_3$ we obtain that $\hat{n} \in (1 \pm \gamma)n_H(G_\theta)$. In particular, it holds that $n_H(G_\theta) \geq \epsilon m/2$. Hence conditioned on $E_1 \cap E_2 \cap E_3$ we obtain that Equation 4 holds in this case as well. In particular $\hat{n}\hat{Z} \geq (1 - \gamma)^2 \frac{\beta m}{|E(H)|} > \epsilon m(1 + \gamma)^2$ (since $(1 + \eta) > (1 + 2\gamma)^5 > (1 + \gamma)^2/(1 - \gamma)^3$). Thus, conditioned on $E_1 \cap E_2 \cap E_3$, the algorithm rejects G , as desired. ◀

▷ **Claim 16.** The expected query complexity of Tolerant-Test-Subgraph-freeness is bounded above by the query complexity of performing $\ell = \Theta(\Delta/\eta^2)$ random-copy queries to $(G_\theta)^H$ with parameters $s = \epsilon m/2$ and $\delta = \Theta(\eta^2/\Delta)$ (see Definition 7), $\Theta(\ell \cdot \Delta)$ all-neighbor queries to $(G_\theta)^H$, where $\Delta = \Delta((G_\theta)^H)$, and a single execution of a H -copies number estimator for G_θ (see Definition 8).

Proof. By construction, linearity of expectation and Theorem 11. ◁

4 The oracles for k -cliques

In this section we describe our K_k -copies oracle and K_k -copies number estimator. We conclude with our main theorem about tolerant testing of K_k -freeness (Theorem 21).

▷ **Claim 17.** For every iteration $j \in [t]$ of Algorithm 4 it holds that:

$$\mathbb{E}(Y_j) = n_{K_k}(G_\theta) / \left(n \cdot \binom{\theta}{k-1} \right). \quad (5)$$

Proof. Consider a copy, \mathcal{K} , of K_k in G_θ . Let $y(\mathcal{K})$ denote the number of θ -light vertices in \mathcal{K} . The probability that \mathcal{K} is found in Step 2f of Approximate- K_k -copies-in- G_θ is exactly $y(\mathcal{K}) \cdot 1/(n \binom{\theta}{k-1})$. Conditioned on the event that indeed \mathcal{K} was found in Step 2f of iteration j the probability that Y_j is set to be 1 is $1/y(\mathcal{K})$. Thus, for every j , $\Pr(Y_j = 1) = n_{K_k}(G_\theta)/(n \binom{\theta}{k-1})$. The claim follows. ◁

▷ **Claim 18.** Approximate- K_k -copies-in- G_θ is a K_k -copies number estimator of the graph G_θ . Its query complexity is $O(k^2 n \binom{\theta}{k-1} / (\gamma^2 s))$.

Algorithm 4 Approximate- K_k -copies-in- G_θ .

Input: $\theta, s, \gamma \in (0, 1/2]$

1. Set $t = \Theta(n \binom{\theta}{k-1} / (\gamma^2 s))$
 2. For $j = 1, \dots, t$
 - a. Sample a vertex $v \in V$ u.a.r. and a set of $k-1$ indexes, i_1, \dots, i_{k-1} u.a.r. from $\binom{[\theta]}{k-1}$.
 - b. If v θ -heavy then set $Y_j = 0$ and go to the next iteration.
 - c. Otherwise, query (v, i_ℓ) for each $\ell \in [k-1]$. Let u_1, \dots, u_{k-1} denote the respective return values.
 - d. Let y denote the number of θ -light vertices in $\{u_i\}_{i \in [k-1]}$.
 - e. If $y < k-2$ then set $Y_j = 0$ (there is more than a single θ -heavy vertex in the set).
 - f. If the subgraph induced on $\{u_i\}_{i \in [k-1]}$ is a $(k-1)$ -clique then set $Y_j = 1$ w.p. $1/(y+1)$. Otherwise set $Y_j = 0$.
 3. Return $n \cdot \binom{\theta}{k-1} \cdot \sum_{j \in [t]} Y_j / t$.
-

Proof. For $s \leq n_{K_k}(G_\theta)$ the claim follows from Claim 16, the setting of t and multiplicative Chernoff's bound (see Theorem 30). To see the correctness of the second part, consider $s = n_{K_k}(G_\theta)$. By the first part of the claim, w.h.c.p. the return value of the algorithm is in $(1 \pm \gamma)n_{K_k}(G_\theta) = (1 \pm \gamma)s$. Namely, at most $(1 + \gamma)s$. Now consider the case in which $s > n_{K_k}$. By a coupling argument, w.h.c.p. the return value is at most $(1 + \gamma)s$ in this case. To see this, couple the return value of the algorithm to the return value of the algorithm on the graph after we add $s - n_{K_k}$ copies to the graph (arbitrarily). Since we added copies, the return value of the algorithm on the modified graph dominates the return value on the original graph (to see this consider executing the algorithm in parallel in both graphs - the outcome in the modified graph is always at least as high) but is guaranteed to be at most $(1 + \gamma)s$ w.h.c.p. by the above. The claim about the query complexity follows from construction. This concludes the proof. \triangleleft

Algorithm 5 Get-Random- K_k -Copy-in- G_θ .

Input: θ, s, δ

1. Repeat $\Theta(n \binom{\theta}{k-1} \log(1/\delta)/s)$ times:
 - a. Sample a vertex $v \in V$ u.a.r. and a set of $k-1$ indexes, i_1, \dots, i_{k-1} u.a.r. from $\binom{[\theta]}{k-1}$.
 - b. If v is not θ -light then break.
 - c. Otherwise, query (v, i_ℓ) for each $\ell \in [k-1]$. Let u_1, \dots, u_{k-1} denote the respective return values.
 - d. Let y denote the number of θ -light vertices in $\{u_i\}_{i \in [k-1]}$.
 - e. If $y < k-2$ or if the subgraph induced on $\{u_i\}_{i \in [k-1]}$ is not a clique, then go to the next iteration.
 - f. Otherwise, w.p. $1/(y+1)$, return the k -clique induced on v and $\{u_i\}_{i \in [k-1]}$.
-

\triangleright Claim 19. The query complexity of Get-Random- K_k -Copy-in- G_θ is $\Theta(k^2 n \binom{\theta}{k-1} \log(1/\delta)/s)$. If $s < n_{K_k}(G_\theta)$ then it returns a copy w.p. at least $1 - \delta$.

Proof. For any copy of K_k , \mathcal{K} , in each iteration of the while-loop, the probability that we return \mathcal{K} in Step 1f of the algorithm is exactly $1/(n \binom{\theta}{k-1})$. Therefore if the number of copies is greater than s then the probability that it will return a copy during in a single iteration is at least $s/(n \binom{\theta}{k-1})$. The claim follows by the setting of the number of iterations. \triangleleft

Algorithm 6 K_k -All-Neighbors-in- G_θ .

Input: θ, \mathcal{K}

1. Set $C = \emptyset$
 2. If $k = 3$, reveal the neighbors of all the θ -light vertices of \mathcal{K} . Let S denote the set of revealed neighbors. If there is a θ -heavy vertex, u , in \mathcal{K} , perform a pair query between u and each one of the vertices in S .
 3. Otherwise, for each θ -light vertex, v , of \mathcal{K} : Reveal all the neighbors of v in G and then for each θ -light neighbor, reveal all its neighbors.
 4. Add all the K_k -copies that were revealed in the previous steps that belong to G_θ (namely, that have at most one θ -heavy vertex) and share an edge with \mathcal{K} .
 5. Return C .
-

▷ **Claim 20.** The query complexity of K_k -All-Neighbors-in- G_θ is $O(k\theta)$ for $k = 3$ and $O(k\theta^2)$ for $k > 3$.

Proof. The claim follows from construction. ◁

► **Theorem 21.** *There exists a tolerant tester for the property of being K_k -free that distinguishes graphs that are ϵ -close to being K_k free from graphs that are $\epsilon \binom{k}{2} (1 + \eta)$ -far from being K_k -free, where $\epsilon \in (0, 1]$ and $\eta \in (0, 1]$ are parameters. The expected query complexity of the algorithm is $\tilde{O}\left(\frac{k^4 \theta^{2k-3}}{\eta^2 \epsilon d} + k^5 \theta^{2k-4} \theta^{\min\{2, k-2\}}\right)$ where $\theta = \Theta(\text{arb}(G)/(\eta\epsilon))$.*

Proof. The theorem follows from the fact that $\Delta((G_\theta)^H) \leq \binom{k}{2} \theta^{k-2}$, Theorem 15 and claims 16-20. ◀

5 The oracles for 2-connected motifs of radius 1

In this section, H is a motif which is 2-connected and has radius 1. We make the following observation on the copies of H in G_θ .

► **Observation 22.** *For any copy of H , \mathcal{H} , in G_θ , at least one of the following holds:*

1. *there exists a θ -light vertex in \mathcal{H} which is incident to all the vertices in \mathcal{H} .*
2. *there is exactly one θ -heavy vertex in \mathcal{H} .*

Proof. Let v be a vertex in \mathcal{H} which is incident to all other vertices in \mathcal{H} (such vertex exists since H has radius 1). If v is θ -light then Item 1 holds. Otherwise, v is θ -heavy. Since there are no edges in G_θ such that both endpoints are θ -heavy, the claim follows. ◀

We say that a BFS is a θ -restricted if it explores only the neighbors of the θ -light vertices it reaches. We say that an exploration *reveals a copy* \mathcal{H} if all the vertices of \mathcal{H} appeared in the exploration.

▷ **Claim 23.** Let H be a motif that is 2-connected and has radius 1. In every copy of H in G_θ , \mathcal{H} , for every θ -light vertex v in \mathcal{H} , a θ -restricted BFS from v of depth $|V(H)|$ reveals \mathcal{H} .

Proof. If item 1 of observation 22 holds, then clearly the claim follows. Otherwise, there is exactly one θ -heavy vertex, u , in \mathcal{H} . Since H is 2-connected, after we remove u from \mathcal{H} , \mathcal{H} remains connected and its diameter is at most $|V(H)| - 1$. Thus, a θ -restricted BFS of depth $|V(H)| - 1$ from any θ -light vertex of \mathcal{H} reveals all the θ -light vertices of \mathcal{H} . The claim follows from the fact that at least one of the θ -light vertices in \mathcal{H} is a neighbor of u . ◁

Therefore, by Claim 23, finding for a θ -light vertex, v , all the H -copies in G_θ that include v , can be done by using $O(\theta^{|V(H)|})$ queries. Consequently, a query for all-neighbors in $(G_\theta)^H$ can be answered by making $O(|V(H)|\theta^{|V(H)|})$ queries to G . This can be achieved by going over all vertices of the queried copy, \mathcal{H} , and for each θ -light vertex, finding all its copies. From the union of these sets we then remove all the copies that do not share an edge with \mathcal{H} and obtain all the neighbors of \mathcal{H} in $(G_\theta)^H$.

To support random copy queries we assign each copy of H to its θ -light vertex of lowest id. To sample a H -copy u.a.r. from G_θ we first pick a vertex $v \in V$ u.a.r., if it is θ -light then we find all its H -copies as described above. We then pick a copy that is assigned to v u.a.r. and return it w.p. $a_{H,\theta}(v)/a_{H,\theta}^{max}$ where $a_{H,\theta}(v)$ denotes the number of copies that are assigned to v in G_θ and $a_{H,\theta}^{max}$ denotes the maximum number of copies that include a specific θ -light vertex (recall that the copies in G_θ are assigned only to θ -light vertices). Clearly, the above algorithm samples each copy of H in G_θ w.p. $1/(n \cdot a_{H,\theta}^{max})$. Consequently, the probability that it returns a copy is $n_H(G_\theta)/(n \cdot a_{H,\theta}^{max})$. Thus, the success probability of the algorithm depends on $a_{H,\theta}^{max}$. A straightforward upper-bound on $a_{H,\theta}^{max}$ is $|V(H)|! \binom{\theta^{|V(H)|}}{|V(H)|}$ where the first term is due to all possible labeling of the vertices and the second term is due to all possible ways to pick the vertices of the copy from the set of vertices we explored. In the next claim we provide a tighter bound on $a_{H,\theta}^{max}$.

▷ **Claim 24.** For any motif H , it holds that $a_{H,\theta}^{max} \leq |V(H)| \cdot \theta^{|V(H)|-1}$.

Proof. Fix an arbitrary labeling of H . Let $v \in V$. We next bound $a_{H,\theta}(v)$. Label the H -copies that v belongs to according to the fixed labeling of H (each copy has its own labeling). If there is more than one such labeling per copy (due to automorphism), pick one arbitrarily. We next describe a one-to-one mapping from the set of labeled copies to a set of sequences of integers. We then bound the cardinality of the latter set and obtain a bound on the number of H -copies that v belongs to and hence a bound on $a_{H,\theta}^{max}$. Fix a labeled copy of v , \mathcal{H} , and consider the BFS exploration of \mathcal{H} starting at v , where vertices with smaller label are explored first. Given the label of v and the sequence of the vertices in \mathcal{H} by their BFS order we can reconstruct the labels of all the vertices in the copy. The set of vertices and their labels are mapped to a single copy. Moreover, by the above, we can identify the copy by the identity of the root, its label and a sequence of $|V(H)| - 1$ indexes in $[\theta]$. Each index indicates the index of the respective element in its parent adjacency-list. The indexes are listed according to the BFS exploration. Since the identity of the root is known, we can inductively reconstruct the tree by the sequence of indexes. Note that the parents are always θ -light (while the leaves may be θ -heavy) therefore it suffices to use indexes in $[\theta]$. From the above we see that for a specific label of v there are at most $\theta^{|V(H)|-1}$ copies of H to which v belongs. Since there are $|V(H)|$ possible labels, we get the desired bound. ◁

▷ **Claim 25.** *Approximate- H -copies-in- G_θ* is a H -copies number estimator of the graph G_θ . Its query complexity is $O(n \cdot a_{H,\theta}^{max} \cdot \theta^{|V(H)|} / (\gamma^2 s))$.

Proof. The claim about the query complexity follows by construction. Consider the random variable Y_j which is defined in *Approximate- H -copies-in- G_θ* . Since $\mathbb{E}(Y_j) \geq n_H(G_\theta)/n$ and $Y_j \leq a_{H,\theta}^{max}$, for $s \leq n_H(G_\theta)$, by the setting of t and the multiplicative Chernoff's bound (Theorem 30), w.h.c.p the output of the algorithm is in $(1 \pm \gamma)n_H(G)$, as required. For $s > n_H(G_\theta)$, by a coupling argument, w.h.c.p. the output of the algorithm is at most $(1 + \gamma)$ (see more details in the proof of Claim 14). ◁

Algorithm 7 Approximate- H -copies-in- G_θ .

Input: $s, \gamma \in (0, 1/2], \theta$

1. Set $t = \Theta(n \cdot a_{H,\theta}^{max} / (\gamma^2 s))$
 2. For $j = 1, \dots, t$
 - a. Sample a vertex $v \in V$ u.a.r.
 - b. If v is not θ -light then set $Y_j = 0$ and go to the next iteration.
 - c. Otherwise, perform a θ -restricted BFS from v to depth $|V(H)|$ and reveal all the H -copies assigned to v in G_θ . Set Y_j to be this number.
 3. Return $n \cdot \sum_{j \in [t]} Y_j / t$.
-

Algorithm 8 Get-Random- H -Copy-in- G_θ .

Input: θ, s, δ

1. Repeat $\Theta(n \cdot a_{H,\theta}^{max} \log(1/\delta) / s)$ times:
 - a. Sample a vertex $v \in V$ u.a.r.
 - b. If v is θ -heavy then go to the next iteration.
 - c. Otherwise, perform a θ -restricted BFS from v to depth $|V(H)|$ and reveal all the H -copies assigned to v in G_θ .
 - d. Pick a random copy from the H -copies assigned to v and return it with probability $a_{H,\theta}(v) / a_{H,\theta}^{max}$.
-

▷ **Claim 26.** The query complexity of **Get-Random- H -Copy-in- G_θ** is $\Theta(\theta^{|V(H)|} \cdot n \cdot a_{H,\theta}^{max} \log(1/\delta) / s)$. It returns a H -copy w.p. at least $1 - \delta$.

Proof. The claim about the query complexity follows by construction. In each iteration of the algorithm, each copy of H is returned with probability $1/(n \cdot a_{H,\theta}^{max})$. Thus, in each iteration a copy is returned w.p. $n_H(G_\theta) / (n \cdot a_{H,\theta}^{max})$. Thus, if $s \leq n_H(G)$, a copy is returned w.p. at least $1 - \delta$ by the setting of the number of iterations. ◁

Algorithm 9 H -All-Neighbors-in- G_θ .

Input: θ

1. Set $C = \emptyset$
 2. For each vertex v of \mathcal{H} :
 - a. Reveal all the H -copies of v in G^θ by performing a θ -restricted BFS to depth $|V(H)|$ from v in G .
 - b. Add to C all the revealed H -copies that share an edge with \mathcal{H} .
 3. Return C .
-

▷ **Claim 27.** The query complexity of K_k -All-Neighbors-in- G_θ is $O(|V(H)|\theta^{|V(H)|})$.

Proof. By Construction. ◁

▶ **Theorem 28.** *There exists a tolerant tester for the property of being H -free for H that is 2-connected and has radius 1 that distinguishes graphs that are ϵ -close to being H free from graphs that are $\epsilon|E(H)|(1+\eta)$ -far from being H -free, where $\epsilon \in (0, 1]$ and $\eta \in (0, 1]$ are parameters. The expected query complexity of the algorithm is $\tilde{O}\left((h^7/\eta^2 + h^3/(\eta\epsilon)) \cdot \theta^{3h-2}\right)$ where $\theta = \Theta(\text{arb}(G)/(\eta\epsilon))$ and $h = |V(H)|$.*

Proof. The theorem follows from the fact that $\Delta((G_\theta)^H) \leq h \cdot a_{H,\theta}^{max}$, Theorem 15 and claims 24–27. \blacktriangleleft

6 The tester for the bounded-degree model

Clearly, Tolerant-Test-Subgraph-freeness applies also to bounded degree graphs. Moreover, for the setting where $\theta = d + 1$ the graph G_θ is simply G and all the vertices in the graph are θ -light. Consequently, we can test H -freeness for *any* motif H by using the oracles in Section 5 without making any modifications ². We obtain the following result.

► **Theorem 29.** *There exists a tolerant tester, in the bounded degree model, for the property of being H -free for any motif H . Specifically the tester distinguishes graphs that are ϵ -close to being H free from graphs that are $\epsilon|E(H)|(1 + \eta)$ -far from being H -free, where $\epsilon \in (0, 1]$ and $\eta \in (0, 1]$ are parameters. The expected query complexity of the algorithm is $\tilde{O}((h^7/\eta^2 + h^3/(\eta\epsilon d)) \cdot d^{3h-2})$ where $h = |V(H)|$ and d is the degree bound.*

Proof. By the proof of Theorem 21. \blacktriangleleft

References

- 1 Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. Testing triangle-freeness in general graphs. *SIAM Journal on Discrete Mathematics*, 22(2):786–819, 2008. doi:10.1137/07067917X.
- 2 Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 93–102. IEEE Computer Society, 2002. doi:10.1109/SFCS.2002.1181886.
- 3 Talya Eden, Reut Levi, and Dana Ron. Testing c_k -freeness in bounded-arboricity graphs. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 60:1–60:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.60.
- 4 Talya Eden, Dana Ron, and Will Rosenbaum. The arboricity captures the complexity of sampling edges. In *46th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 52:1–52:14, 2019. doi:10.4230/LIPICS.ICALP.2019.52.
- 5 Talya Eden, Dana Ron, and Will Rosenbaum. Almost optimal bounds for sublinear-time sampling of k -cliques in bounded arboricity graphs. In *49th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 56:1–56:19, 2022. doi:10.4230/LIPICS.ICALP.2022.56.
- 6 Talya Eden, Dana Ron, and C. Seshadhri. Sublinear time estimation of degree distribution moments: The arboricity connection. *SIAM J. Discret. Math.*, 33(4):2267–2285, 2019. doi:10.1137/17M1159014.
- 7 Talya Eden, Dana Ron, and C. Seshadhri. Faster sublinear approximation of the number of k -cliques in low-arboricity graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1467–1478, 2020. doi:10.1137/1.9781611975994.89.
- 8 Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. *Theory Comput.*, 2(9):173–183, 2006. doi:10.4086/TOC.2006.V002A009.
- 9 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/S00453-001-0078-7.

² In fact the query complexity of Algorithms 7–9 can be slightly improved by setting the depth of the BFS exploration to be the diameter of H (instead of $|V(H)|$).

- 10 L. Gugelmann. Testing triangle-freeness in general graphs: Lower bounds. Bachelor thesis, Dept. of Mathematics, ETH, Zurich, 2006.
- 11 Tali Kaufman, Michael Krivelevich, and Dana Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, 2004. doi:10.1137/S0097539703436424.
- 12 Reut Levi. Testing triangle freeness in the general model in graphs with arboricity $o(\sqrt{n})$. In *48th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 198, pages 93:1–93:13, 2021. doi:10.4230/LIPICS.ICALP.2021.93.
- 13 Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Trans. Algorithms*, 5(2):22:1–22:28, 2009. doi:10.1145/1497290.1497298.
- 14 Michal Parnas and Dana Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, 20(2):165–183, 2002. doi:10.1002/RSA.10013.
- 15 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.
- 16 T. Rast. Testing triangle-freeness in general graphs: Upper bounds. Bachelor thesis, Dept. of Mathematics, ETH, Zurich, 2006.
- 17 Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. Improved constant-time approximation algorithms for maximum matchings and other optimization problems. *SIAM J. Comput.*, 41(4):1074–1093, 2012. doi:10.1137/110828691.

A

 Appendix

► **Theorem 30** (Multiplicative Chernoff’s Bound). *Let X_1, \dots, X_n be identical independent random variables ranging in $[0, 1]$, and let $p = \mathbb{E}[X_1]$. Then, for every $\gamma \in (0, 2]$, it holds that*

$$\Pr \left[\left| \frac{1}{n} \cdot \sum_{i \in [n]} X_i - p \right| > \gamma \cdot p \right] < 2 \cdot e^{-\gamma^2 pn/4} . \quad (6)$$