An Optimal Separation Between Two Property Testing Models for Bounded Degree Directed Graphs

Pan Peng
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Yuyang Wang
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Abstract

We revisit the relation between two fundamental property testing models for bounded-degree *directed* graphs: the *bidirectional* model in which the algorithms are allowed to query both the outgoing edges and incoming edges of a vertex, and the *unidirectional* model in which only queries to the outgoing edges are allowed. Czumaj, Peng and Sohler [STOC 2016] showed that for directed graphs with both maximum indegree and maximum outdegree upper bounded by d, any property that can be tested with query complexity $O_{\varepsilon,d}(1)$ in the bidirectional model can be tested with $n^{1-\Omega_{\varepsilon,d}(1)}$ queries in the unidirectional model. In particular, if the proximity parameter ε approaches 0, then the query complexity of the transformed tester in the unidirectional model approaches n. It was left open if this transformation can be further improved or there exists any property that exhibits such an extreme separation.

We prove that testing *subgraph-freeness* in which the subgraph contains k source components, requires $\Omega(n^{1-\frac{1}{k}})$ queries in the unidirectional model. This directly gives the first explicit properties that exhibit an $O_{\varepsilon,d}(1)$ vs $\Omega(n^{1-f(\varepsilon,d)})$ separation of the query complexities between the bidirectional model and unidirectional model, where $f(\varepsilon, d)$ is a function that approaches 0 as ε approaches 0. Furthermore, our lower bound also resolves a conjecture by Hellweg and Sohler [ESA 2012] on the query complexity of testing k-star-freeness.

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

Keywords and phrases Graph property testing, Directed graphs, Lower bound, Subgraph-freeness

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.96

Category Track A: Algorithms, Complexity and Games

Funding This work has been supported in part by NSFC grant 62272431 and "the Fundamental Research Funds for the Central Universities".

1 Introduction

Graph property testing is a framework for studying extremely fast (randomized) algorithms for solving a relaxation of classical decision problems on graphs. Given a graph property P, we are interested in designing an algorithm, called a *property tester*, that with high constant probability, accepts any graph G that satisfies P, and rejects any graph that is "far" from satisfying P, i.e., one needs to modify a significant fraction of the representation (e.g., adjacency matrix or adjacency list) of the graph to make it satisfy P. It is assumed that the algorithm is given oracle access to the representation of the graph and the goal of a property tester is to solve the above problem by making as few queries to the oracle as possible. Since the seminal works by Rubinfeld and Sudan [20] (on algebraic property testing) and Goldreich, Goldwasser and Ron [11] (on combinatorial and graph property testing), a lot of efforts have been made on studying which properties can be tested within a sublinear (e.g., constant) number of queries in several classical models, e.g., the dense graph model [11, 1] and bounded-degree graph model [12]. In particular, we have see a rapid development of property testing on *undirected* graphs in the past two decades. We refer to the recent book [10] for a survey.

On the other hand, we still do not know much about property testing in *directed graphs (digraphs)* so far. Bender and Ron [3] introduced two fundamental models for studying directed graph property testing. The first is called *bidirectional* model, where the algorithm is allowed to query both outgoing and incoming edges of a vertex; the second is called *unidirectional* model, where the algorithm is only allowed to query the outgoing edges, while not incoming edges. The latter model seems more realistic for some applications. For example, consider the webgraphs. It is much easier to query the outgoing edges (u, v) (which corresponds to a hyperlink from webpage u to webpage v) than querying the incoming edges. In this paper, we focus on bounded-degree directed graphs. A digraph G is said to be *d-bounded*, if both the maximum outdegree and maximum indegree of G are upper bounded by d, which is assumed to be a constant.

Bender and Ron gave an algorithm for testing strong connectivity with $\tilde{O}(1/\varepsilon)$ queries in the bidirectional model, and showed that there is a lower bound of $\Omega(\sqrt{n})$ queries for any algorithm with two-sided error[1] in the unidirectional model. Goldreich [9], and Hellweg and Sohler [16] gave a lower bound of $\Omega(n)$ queries for testing strong connectivity with one-sided error in the unidirectional model. The works [9, 16] also gave testers for strong connectivity with $n^{1-1/(d+1/\varepsilon)}$ queries with two-sided error in the unidirectional model. In [16], the authors gave testers for subgraph-freeness with $O(n^{1-\frac{1}{k}})$ queries in the unidirectional model, where k is the number of connected components in the subgraph that have no incoming edges. It is known that a few properties can be tested with a constant number of queries in the bidirectional model, including Eulerianity [18], k-edge connectivity [18, 23, 8], k-vertex connectivity [18, 8].

Towards a deeper understanding of testing properties of bounded degree directed graphs (digraphs), Czumaj, Peng and Sohler [5] studied the relation between these two models and provided a generic transformation that converts testers with constant query complexity in the bidirectional model, to testers with sublinear query complexity in the unidirectional model. Specifically, in [5], it was shown that any property P that can be tested with[2] $q = O_{\varepsilon,d}(1)$ queries in the bidirectional model can be tested with $n^{1-d^{-\Theta(q)}} = n^{1-\Omega_{\varepsilon,d}(1)}$ queries in the unidirectional model (with two-sided error). In particular, if the proximity parameter ε approaches 0, then the query complexity of the transformed tester in the unidirectional model approaches n (as the term $d^{-\Theta(q)}$ approaches 0).

One natural question that is left open is that *is the above the transformation tight*? Or equivalently, *can we achieve a much better query complexity, say $n^{c-\Omega_{\varepsilon,d}(1)}$, in the latter model, for some universal constant $c < 1$*? Indeed, currently, the best known lower bound for this transformation is for testing 3-*star-freeness*, where a 3-star is a 4-vertex directed graph such that there exists one center vertex v, and for any other three vertices u, there is an edge from u to v, and no other edges exist. Hellweg and Sohler [16] have shown that 3-star-freeness can be tested with a constant number of queries in bidirectional model, while

[1] A tester for a property P is said to have *one-sided error* if it accepts every (di)graph satisfying P, and it errs if the graph is far from having P. It is said to have *two-sided error* if it errs in both cases.

[2] Throughout the paper, we use the notation $O_{\varepsilon,d}()$ (resp. $\Omega_{\varepsilon,d}()$) to describe a function in the Big-O (resp. Big-Omega) notation assuming that ε and d are constant.

the query complexity of a tester for this property in the unidirectional model is $\Theta(n^{2/3})$ for any constant $\varepsilon > 0$. Therefore, there is still a significant gap between the upper bound (i.e., $n^{1-\Omega_{\varepsilon,d}(1)}$) in the bidirectional model in the transformation and the current best-known lower bound (i.e., $\Omega(n^{2/3})$).

Before we state our result, we formally introduce the definition of property testing in both directional and unidirectional models. Let $P = (P_n)_{n\in\mathbb{N}}$ be a d-bounded digraph property, where P_n is a property of d-bounded digraphs with n vertices. An n-vertex graph G is said to be ε-far from satisfying P_n if one needs to modify more than εdn edges to make it a d-bounded digraph with property P_n, where $\varepsilon > 0$ is called the proximity parameter. We say that P is *q-query testable* (or that P *can be tested with query complexity* q) if for every n, ε and d, there exists a tester that makes $q = q(n, \varepsilon, d)$ queries and with probability at least $\frac{2}{3}$, accepts any n-vertex d-bounded digraph G satisfying P, and rejects any n-vertex d-bounded digraph G that is ε-far from satisfying P. We call such a tester an *ε-tester* for P.

We show that there exists a property that exhibits an $O_{\varepsilon,d}(1)$ vs $\Omega(n^{1-\Theta_{\varepsilon,d}(1)})$ separation of the query complexities between the bidirectional model and unidirectional model, which implies that the transformation of [5] is essentially tight.

▶ **Theorem 1.** *For any sufficiently small constant $\varepsilon > 0$, there exists a digraph property $P = P_{\varepsilon,d}$ such that P can be tested with $O_{\varepsilon,d}(1)$ queries in the bidirectional model, while any ε-tester for P in the unidirectional model requires $n^{1-f(\varepsilon,d)}$ queries, where $f(\varepsilon, d)$ is a function that approaches 0 when ε approaches 0.*

The above theorem is a direct corollary from the following result regarding testing subgraph-freeness. Let H be a directed graph. A strongly connected component[3] W is called a *source component* of H, if there is no edge from $V(H) \setminus W$ to W. A directed graph H is said to be *weakly connected* if its underlying undirected graph (i.e., the graph that is obtained by ignoring all the directions of the edges) is connected. For example, we note that k-star is just a weakly connected digraph with k source components, where a directed graph H with $k + 1$ vertices is called a *k-star* if there is a vertex v such that each of the other k vertices has exactly one edge pointing to v, and H does not contain any other edges. Let G and H be two directed graphs. The graph G is said to be *H-free* if H does not appear as a subgraph in G. We have the following theorem on testing H-freeness for any (constant-size) H with k source components.

▶ **Theorem 2.** *Let k be any integer such that $k \geq 2$. Let d be some constant. Let H be a weakly connected*[4] *directed graph with k source components. There exists an $\varepsilon_0 = \Theta_{d,k}(1)$ such that any ε_0-tester for testing H-freeness of an n-vertex d-bounded graph requires at least $\Omega(n^{1-\frac{1}{k}})$ queries in the unidirectional model.*

We remark that it has been shown by Hellweg and Sohler [16] that for any H with k source components, H-freeness can be tested with query complexity $O_{\varepsilon,d,k}(1)$ in the bidirectional model, and also can be tested with query complexity $O_{\varepsilon,d,k}(n^{1-\frac{1}{k}})$ in the unidirectional model[5]. Given the above result, we can easily prove Theorem 1.

[3] We call $W \subseteq V(H)$ a *strongly connected component* of H if the subgraph $H[W]$ of H induced by W is strongly connected, and there does not exist any set of vertices $X \subseteq V(H) \setminus W$ such that the subgraph of H induced by $W \cup X$ is strongly connected. That is, the subgraph of $H[W]$ is a strongly connected and maximal.

[4] For graphs H that is not weakly connected, we can handle each of its weakly connected components separately.

[5] On the high level, their algorithms use the following observation: if a bounded-degree directed graph G

Proof of Theorem 1. Let $\eta > 0$ and define property P_η to be the property of being H-free, for any H that is weakly connected digraph with $k = \lceil 1/\eta \rceil$ source components. According to Theorem 2, any ε_0-tester for P_η requires at least $\Omega(n^{1-\frac{1}{k}}) > \Omega(n^{1-\eta})$ queries in the unidirectional model, where $\varepsilon_0 = \varepsilon_0(d, \eta)$ is a function of d, η. Now given any sufficiently small constant $\varepsilon > 0$, let η' be a number satisfying that $\varepsilon = \varepsilon_0(d, \eta')$. Then Theorem 1 follows by taking $P = P_{\eta'}$ and $f(\varepsilon, d) = \eta'$. ◀

Furthermore, it was conjectured in [16] that testing k-star-freeness requires $\Omega(n^{1-\frac{1}{k}})$ queries in the unidirectional model. Since k-star is a directed subgraph with k-source components, our Theorem 2 resolves this conjecture.

1.1 Discussions of previous ideas and our techniques

We first sketch the main ideas of the lower bound for testing 3-star-freeness given by Hellweg and Sohler [16]. Their proof makes use of a problem called *testing* 3*-occurrence-freeness*[6] of a sequence[7]. Let A be a length-n sequence of integers such that each element in A is from $[\ell] := \{1, \cdots, \ell\}$ and occurs at most 3 times. We say A is 3*-occurrence-free* if no integer in A occurs exactly 3 times in A. We say A is *ε-far from being* 3*-occurrence-free* if one needs to change[8] more than εn elements of A to obtain a 3-occurrence-free sequence. [16] gave a local reduction from the problem of testing 3-occurrence-freeness of a sequence to the problem of testing 3-star-freeness. That is, given an instance A with m elements of 3-occurrence-freeness, they constructed a graph G with $\Theta(m)$ vertices, such that

1) if A is 3-occurrence-free, then G is 3-star-free; if A is ε-far from being 3-occurrence-free then G is $\Theta(\varepsilon)$-far from being 3-star-free;
2) every query to G can be answered by performing $O(1)$ queries to A.

To obtain a lower bound for testing 3-occurrence-freeness, [16] constructed two classes $\mathcal{C}_A$, $\mathcal{C}_B$ of length-n sequences such that $\mathcal{C}_A$ is a class of 3-occurrence-free sequences and $\mathcal{C}_B$ is a class of sequences that are $\Omega(1)$-far from being 3-occurrence-free, and the *frequency variables*, denoted by X_A and X_B, of the sequences from these two different classes have 2 proportional moments, i.e.,

$$\frac{\mathbf{E}[X_B]}{\mathbf{E}[X_A]} = \frac{\mathbf{E}[X_B^2]}{\mathbf{E}[X_A^2]}.$$

Then the lower bound $\Omega(n^{2/3})$ for testing 3-occurrence-freeness follows from a lower bound for distinguishing random variables with 2-proportional moments given in [19].

Now we note that to obtain a lower bound for testing H-freeness for any H with k source components, it suffices to give a lower bound for testing k-occurrence-freeness for general k in the way similar as above. That is, we construct two classes $\mathcal{C}_A$, $\mathcal{C}_B$ of length-n sequences such that $\mathcal{C}_A$ is a class of k-occurrence-free sequences and $\mathcal{C}_B$ is a class of sequences that are $\Omega_k(1)$-far from being k-occurrence-free, and the *frequency variables*, denoted by X_A and X_B, of the sequences from these two different classes have $k-1$ proportional moments, i.e.,

$$\frac{\mathbf{E}[X_B]}{\mathbf{E}[X_A]} = \frac{\mathbf{E}[X_B^2]}{\mathbf{E}[X_A^2]} = \cdots = \frac{\mathbf{E}[X_B^{k-1}]}{\mathbf{E}[X_A^{k-1}]}.$$

is ε-far from H-freeness, then G contains $\Omega(\varepsilon n)$ vertex-disjoint copies of H. Then in the bidirectional model, one can sample a constant number of vertices and perform BFS from each sampled vertex to find a copy of H; in the unidirectional model, one can sample many edges to see if some copy of H is formed.

[6] In [16], the same problem was called 3-value freeness.

[7] We use "sequence" rather than "multiset" as the position of each element affects our construction.

[8] It is allowed to use integers that are larger than ℓ to change the elements of A.

However, the main difficulty is to construct two classes of sequences satisfying the above equations for general $k \geq 3$, which was also pointed out in [16]. Besides the aforementioned construction in [16] which only works for $k = 3$, we also note that in [19], a special pair of random variables with $k - 1$ proportional moments is also constructed (for establishing their lower bound for DISTINCT-ELEMENTS). That is, their random variables take values of the form $(B + 3)^i$, for any integers $B > 1, k > 1$ and $i = 0, \ldots, k - 1$. This leads to a large gap between the expectations of the corresponding variables. To show a lower bound for testing k-occurrence-freeness, we need to construct random variables taking values $1, 2, \ldots, k$, for any integer $k > 1$. This is more challenging as it corresponds to a much smaller gap (which is arbitrarily close to 1) between the expectations of the corresponding variables (see Lemma 10). To construct such two random variables, we establish some identities related to binomial coefficients, and use them to define two distributions satisfying a number of linear equations which in turn are necessary conditions for two variables having proportional moments.

We then give a local reduction from testing k-occurrence-freeness to testing H-freeness for H with k-source components. The reduction also non-trivially generalizes the one for 3-star-free in [16], as 3-star is a special subgraph with a nice symmetric property, while an arbitrary subgraph H might contain different types of asymmetric structures. Our main idea is as follows. Given a sequence S, we construct a graph G on the fly such that each element in the sequence corresponds to a source component of H in G; an element in S appears k times if and only if a copy of H is added in G. For the latter, we carefully add k source components of H to G and add edges from these components to one center component (which is the rest part of H after removing all the source components). Finally, we show that this construction preserves the distance to the properties and each query to G can be answered by querying at most 1 position in S.

1.2 Other Related work

Ito, Khoury and Newman [17] recently gave a characterization of monotone and hereditary properties that can be tested with constant query complexity and one-sided error in both bounded-degree bidirectional model and bounded-degree unidirectional model. For testing acyclicity in the bidirectional model, Bender and Ron [3] gave a lower bound of $\Omega(n^{1/3})$ queries for algorithms with two-sided error and a lower bound $\Omega(n^{1/2})$ queries for algorithms with one-sided error. The latter lower bound has been improved to $\tilde{\Omega}(n^{5/9})$ queries by Chen, Randolph, Servedio and Sun [4].

In the dense directed graph model (with different types of queries and notion of "ε-far"), Alon and Shapira [2] gave an algorithm with constant query complexity for testing subgraph-freeness.

There exists a class of properties which can be tested with constant number of queries by the so-called *proximity-oblivious testers* [13]. Goldreich and Ron [14] showed that any property that can be tested by a proximity-oblivious tester that makes q uniformly distributed queries with constant detection probability can be tested by a sample-based testers of sample complexity $O(n^{1-1/q})$, where a sample-based tester only samples elements independently from some distribution of the tested object. Building upon [7, 15], Dall'Agnol, Tom and Lachish [6] recently showed that any property that is testable with q queries admits a sample-based tester with sample complexity $n^{1-1/O(q^2 \log^2 q)}$. Their algorithms are defined over a constant-size output alphabet, which is very different from the bounded degree (directed) graph model, in which a super constant alphabet is needed.

Valiant developed a wishful thinking theorem in [22], telling that two distributions whose so-called *k-based moments* have small gap are indistinguishable by k-Poissonized samples. This is a tool for establishing lower bounds of testing symmetric properties on distributions.

On a very high level, both [22] and our work are constructing far distributions with the same collision, while the details for the constructions differ significantly. For example, our proof is built upon Corollary 5.7 of [19], which requires to carefully construct two distributions that have proportional moments. In [22], it is required to construct two distributions whose k-based moments have small gap. It is unclear if two distributions with small gap between k-based moments have proportional moments, or vice versa. In addition, we are using very different properties of Vandermonde matrix from those used in [22].

2 A Lower Bound for Testing k-Occurrence-freeness

In this section, we will prove the lower bound on the query complexity for testing k-occurrence-freeness, which is defined as follows. Given a sequence A of n integers such that each entry of A is from $[n] := \{1, \dots, n\}$ and each element $i \in [n]$ occurs at most k times, the problem is to distinguish if A is *k-occurrence-free*, i.e., no element occurs in k positions of A, or A is *ε-far from k-occurrence-free*, i.e., more than εn elements of A needs to be changed to make it k-occurrence-free. We assume that the algorithm can query the element (or the value) of any position of the sequence in constant time. The goal is to solve the problem by making as few queries as possible. We will show the following result.

▶ **Theorem 3.** *Any algorithm for testing k-occurrence-freeness with parameter $\varepsilon = \Omega_k(1)$ requires at least $\Omega(n^{1-1/k})$ queries, where n is the length of the input sequence.*

2.1 Basic tools and notions

To prove the above theorem, we will make use of a lower bound by Raskhodnikova et al. [19] for distinguishing two sequences satisfying some property. We first introduce two definitions.

▶ **Definition 4** (Frequency variable). *Let A be a sequence of integers. We define its* frequency variable *X_A as follows. Choose a number uniformly at random from the set of distinct elements that occur in A and then let X_A denote its* frequency[9]*, i.e., the number of times it occurs.*

Take the following sequence $S = \{1, 2, 1, 3, 2, 1, 4\}$ as an example. There are 4 distinct elements (or values) in S: value 1 occurs 3 times, value 2 occurs twice, value 3 and 4 each occurs once. Thus the frequency variable X_S of S satisfies that $\Pr[X_S = 1] = 0.5$, $\Pr[X_S = 2] = 0.25$, $\Pr[X_S = 3] = 0.25$.

▶ **Definition 5** (Proportional moments). *Two random variables X_1 and X_2 are said to have* $k-1$ proportional moments, *if $\frac{\mathbf{E}[X_2]}{\mathbf{E}[X_1]} = \frac{\mathbf{E}[X_2^2]}{\mathbf{E}[X_1^2]} = \cdots = \frac{\mathbf{E}[X_2^{k-1}]}{\mathbf{E}[X_1^{k-1}]}$. We say that two sequences have $k-1$ proportional moments if their frequency variables have $k-1$ proportional moments.*

Let P denote a property defined on sequence of integers such that it is invariant under any permutation of indices and values. [19] has shown that any tester for P that makes t queries can be simulated by a *Poisson-s* algorithm that only looks at the histogram of the samples as its input, and $s = O(t)$. Relevant definitions are as follows.

▶ **Definition 6** (*Poisson-s* algorithm). *An algorithm is called a* Poisson-s *algorithm if the number of samples of the algorithm is determined by a Poisson distribution with the expectation s.*

[9] We directly adopt the notion "frequency" from [19].

▶ **Definition 7** (Histogram). *Given a sequence S, the histogram H of S is a function defined as follows:*

$$H(i) := |\{s \in S | s \textit{ occurs exactly } i \textit{ times in } S\}|$$

In [19], Raskhodnikova et al. proved that if two sequences have $k-1$ proportional moments and $s = o(n^{1-\frac{1}{k}})$, then any *Poisson-s* algorithm can't distinguish their histograms.Formally, based on Lemma 5.3 and Corollary 5.7 in [19], we have the following Lemma.

▶ **Lemma 8** ([19]). *Let X_A and X_B be two random variables with $k-1$ proportional moments. And let D_{X_A} and D_{X_B} be two length-n sequences of integers, whose frequency variables are X_A and X_B, respectively. Let P be a property of sequences that is invariant under permutations of indices and values, and let $\varepsilon > 0$ be a constant.*

1. *If $\mathcal{A}'$ is a tester for P with t queries, i.e., $\mathcal{A}'$ accepts the input sequence that satisfies P with probability at least $\frac{2}{3}$; it rejects any sequence that is ε-far from satisfying P, with probability at least $\frac{2}{3}$.*
   Then there must be a Poisson-s *algorithm $\mathcal{A}$ that gets only the histogram of the samples, where $s = O(t)$, satisfiying the following: if the input sequence satisfies P, $\mathcal{A}$ accepts with probability at least $\frac{2}{3} - o(1)$; if the input sequence is ε-far from satisfying P, $\mathcal{A}$ rejects with probability at least $\frac{2}{3} - o(1)$.*
2. *For any* Poisson-s *algorithm $\mathcal{A}$ with $s = o(n^{1-\frac{1}{k}})$, if $\mathcal{A}$ gets only access to the histogram of samples, then we have*

$$|\Pr[\mathcal{A}(D_{X_A}) = \text{True}] - \Pr[\mathcal{A}(D_{X_B}) = \text{True}]| = o(1).$$

Note that by the above Lemma, for a property P that is invariant under permutation of indices and values, any tester for P can be well simulated by a *Poisson-s* algorithm, which only accesses to the histogram of samples. Thus it suffices to only consider such *Poisson-s* algorithms. Furthermore, if there exist two instances of P with proportional moments, then it is hard to distinguish these two instances, for any *Poisson-s* algorithm that only accesses to the histogram of samples.

2.2 Proof of Theorem 3

Now we give the proof of Theorem 3. We first note that k-occurrence-freeness is a property that is invariant under permutation of indices and values. Suppose that there exist two families of sequence instances, denoted by $\mathcal{C}_A$ and $\mathcal{C}_B$, respectively, such that 1) $\mathcal{C}_A$ and $\mathcal{C}_B$ have $k-1$ proportional moments; 2) sequences in $\mathcal{C}_A$ are k-occurrence-free, and sequences in $\mathcal{C}_B$ are far from k-occurrence-freeness. Now assume that there exist a tester $\mathcal{A}'$ for k-occurrence-freeness with $s = o(n^{1-\frac{1}{k}})$ queries. Then, according to Lemma 8, there must be a *Poisson-s* algorithm $\mathcal{A}$ that gets only access to the histogram of samples. For such algorithm $\mathcal{A}$, we have

$$|\Pr[\mathcal{A}(D_{X_A}) = \text{True}] - \Pr[\mathcal{A}(D_{X_B}) = \text{True}]| = (\frac{2}{3} - o(1)) - (\frac{1}{3} + o(1)) \geq \frac{1}{6},$$

which contradicts to the second part of Lemma 8 and thus implies the $\Omega(n^{1-\frac{1}{k}})$ lower bound. Therefore, to prove Theorem 3, it suffices to construct two families of sequences with the above desired properties.

Proof of Theorem 3. We first construct two classes, denoted by $\mathcal{C}_A, \mathcal{C}_B$, of length-$n$ sequences, such that for any sequences $A \in \mathcal{C}_A$ and $B \in \mathcal{C}_B$, it holds that 1) A is k-occurrence-free and B is ε-far from k-occurrence-free, and 2) the frequency variables X_A, X_B of these two instances A, B have $k-1$ proportional moments.

To do so, we first prove the claim.

▷ Claim 9. It holds that

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \cdot \begin{pmatrix} (-1)^1\binom{k}{1} \\ (-1)^2\binom{k}{2} \\ (-1)^3\binom{k}{3} \\ \vdots \\ (-1)^k\binom{k}{k} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{1}$$

Proof. We define a sequence of helper functions $f_j(x)$ to prove (1).

$$f_j(x) = \begin{cases} (1+x)^k, & j = 0 \\ x \cdot f'_{j-1}(x), & j = 1, 2, \cdots, k-1 \end{cases} \tag{2}$$

To prove the claim, we note that it suffices to show the following:

$$f_0(-1) \quad = \quad 1 + \sum_{i=1}^{k} (-1)^{\cdot} \binom{k}{j} = 0, \tag{3}$$

$$f_j(-1) \quad = \quad \sum_{i=1}^{k} i^j \cdot (-1)^i \cdot \binom{k}{i} = 0, \text{ for any } j = 1, \ldots, k-1. \tag{4}$$

Note that if the above are true, then each line of Equations (1) holds, which finishes the proof of the claim. In the following, we prove Equations (3) and (4).

Let us first consider the binomial expansion of $f_0(x)$. We have that

$$f_0(x) = (1+x)^k = \sum_{i=0}^{k} x^i \cdot \binom{k}{i} = 1 + \sum_{i=1}^{k} x^i \cdot \binom{k}{i}. \tag{5}$$

Thus, $f_0(-1) = (1-1)^k = 1 + \sum_{i=1}^{k} (-1)^i \cdot \binom{k}{i} = 0$. That is, Equation (3) holds.

To prove Equation (4), we show that for any $1 \le j \le k-1$, it holds that

(a) $f_j(x) = \sum_{i=1}^{k} i^j \cdot x^i \cdot \binom{k}{i}$,

(b) $f_j(x) = \sum_{i=1}^{j} a_i \cdot x^i \cdot (1+x)^{k-i}$, for some numbers $a_1, \ldots, a_j \ge 0$.

Note that by the above two items, we have that $f_j(-1) = 0 = \sum_{i=1}^{k} i^j \cdot x^i \cdot \binom{k}{i}$, for each $j = 1, \ldots, k-1$, which finishes the proof of Equation (4) (and the claim).

In the following, we prove the above two items (a) and (b) by induction. Consider the case $j = 1$. By definition of function $f_j(x)$ given by (2) and the expansion (5), it holds that

$$f'_0(x) = k \cdot (1+x)^{k-1} = \sum_{i=1}^{k} i \cdot x^{i-1} \cdot \binom{k}{i},$$

which implies that

$$f_1(x) = x \cdot f'_0(x) = x \cdot k \cdot (1+x)^{k-1} = \sum_{i=1}^{k} i \cdot x^i \cdot \binom{k}{i}$$

Now we assume that the items (a) and (b) hold for $j \leq k-2$, and we prove it for $j+1$.

For item (a), since $f_j(x) = \sum_{i=1}^{k} i^j \cdot x^i \cdot \binom{k}{i}$, we have that $f'_j(x) = \sum_{i=1}^{k} i^{j+1} \cdot x^{i-1} \cdot \binom{k}{i}$. Thus,

$$f_{j+1}(x) = \sum_{i=1}^{k} i^{j+1} \cdot x^i \cdot \binom{k}{i}$$

by Definition (2).

For item (b), since $f_j(x) = \sum_{i=1}^{j} a_i \cdot x^i \cdot (1+x)^{k-i}$, for some numbers $a_1, \ldots, a_j \geq 0$, it holds that

$$f'_j(x) = \sum_{i=1}^{j} (a_i \cdot i \cdot x^{i-1} \cdot (1+x)^{k-i} + a_i \cdot x^i \cdot (k-i) \cdot (1+x)^{k-i-1}).$$

Thus, by Definition (2),

$$f_{j+1}(x) = \sum_{i=1}^{j} (a_i \cdot i \cdot x^i \cdot (1+x)^{k-i} + a_i \cdot x^{i+1} \cdot (k-i) \cdot (1+x)^{k-i-1}) = \sum_{i=1}^{j+1} a'_i \cdot x^i \cdot (1+x)^{k-i},$$

for some numbers $a'_1, \cdots, a'_{j+1} \geq 0$.

Therefore, both items (a) and (b) hold and this finishes the proof the claim. ◁

Now we define two distributions $\mathbf{p}, \mathbf{q}$ over $[k]$ as follows.

1. if k is even, define

$$\mathbf{p}_i = \begin{cases} 0, & \text{if } i \text{ is even} \\ \frac{1}{2^{k-1}} \cdot \binom{k}{i}, & \text{if } i \text{ is odd} \end{cases} \qquad \mathbf{q}_i = \begin{cases} \frac{1}{2^{k-1}-1} \cdot \binom{k}{i}, & \text{if } i \text{ is even} \\ 0, & \text{if } i \text{ is odd} \end{cases}$$

2. if k is odd, define

$$\mathbf{p}_i = \begin{cases} \frac{1}{2^{k-1}-1} \cdot \binom{k}{i}, & \text{if } i \text{ is even} \\ 0, & \text{if } i \text{ is odd} \end{cases} \qquad \mathbf{q}_i = \begin{cases} 0, & \text{if } i \text{ is even} \\ \frac{1}{2^{k-1}} \cdot \binom{k}{i}, & \text{if } i \text{ is odd} \end{cases}$$

Now we show the following Lemma.

▶ **Lemma 10.** *Let* $\mathbf{p}, \mathbf{q}$ *be defined as above. There exists* $d > 0$ *such that*

$$\begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \vdots \\ \mathbf{q}_k \end{pmatrix} = d \cdot \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix} + (d-1) \cdot \begin{pmatrix} (-1)^1 \binom{k}{1} \\ (-1)^2 \binom{k}{2} \\ (-1)^3 \binom{k}{3} \\ \vdots \\ (-1)^k \binom{k}{k} \end{pmatrix} \tag{6}$$

Proof. For the case that k is even, we let $d = 1 + \frac{1}{2^{k-1}-1}$.

First note that $\mathbf{p}_k = 0$ and $\mathbf{q}_k = (d-1) \cdot \binom{k}{k}$. Thus, the last equation holds. For even $i \in \{2, 4, \ldots, k\}$, $\mathbf{p}_i = 0$ and $\mathbf{q}_i = d \cdot \mathbf{p}_i + (d-1) \cdot \binom{k}{i}$. For odd $i \in \{1, 3, \ldots, k-1\}$, $\mathbf{p}_i = \frac{d-1}{d} \cdot \binom{k}{i}$ and $\mathbf{q}_i = d \cdot \mathbf{p}_i + (d-1) \cdot (-1) \cdot \binom{k}{i} = 0$. Thus, Equation (6) holds.

For the case that k is odd, we let $d = 1 - \frac{1}{2^{k-1}}$.

Note that $\mathbf{p}_k = 0$ and $\mathbf{q}_k = (1-d) \cdot \binom{k}{k}$. Thus, the last equation holds. For odd $i \in \{1, 3, \ldots, k\}$, $\mathbf{p}_i = 0$ and $\mathbf{q}_i = d \cdot \mathbf{p}_i + (1-d) \cdot \binom{k}{i}$. For even $i \in \{2, 4, \ldots, k-1\}$, $\mathbf{p}_i = \frac{1-d}{d} \cdot \binom{k}{i}$ and $\mathbf{q}_i = d \cdot \mathbf{p}_i + (d-1) \cdot \binom{k}{i} = 0$. Thus, Equation (6) holds. ◀

▶ **Lemma 11.** *Let k be any integer with $k \geq 2$. Let $\mathbf{p}, \mathbf{q}$ be distributions over $[k]$ defined as above. It holds that*

1. $\mathbf{p}_k = 0$ *and* $\mathbf{q}_k \geq \frac{1}{2^k}$*;*
2. *for any two random variables X_A and X_B with distributions $\mathbf{p}$ and $\mathbf{q}$, respectively, it holds that X_A and X_B have $k-1$ proportional moments.*

Proof. The first item follows from the definitions of $\mathbf{p}$ and $\mathbf{q}$.

Now prove the second item. Let $d > 0$ be the number from Lemma 10. We will show that

$$\frac{\mathbf{E}[X_B]}{\mathbf{E}[X_A]} = \frac{\mathbf{E}[X_B^2]}{\mathbf{E}[X_A^2]} = \cdots = \frac{\mathbf{E}[X_B^{k-1}]}{\mathbf{E}[X_A^{k-1}]} = d,$$

or equivalently,

$$\begin{pmatrix} 1 \\ \mathbf{E}[X_B] \\ \mathbf{E}[X_B^2] \\ \vdots \\ \mathbf{E}[X_B^{k-1}] \end{pmatrix} = d \cdot \begin{pmatrix} 1/d \\ \mathbf{E}[X_A] \\ \mathbf{E}[X_A^2] \\ \vdots \\ \mathbf{E}[X_A^{k-1}] \end{pmatrix}. \tag{7}$$

By the definition X_A, it holds that for any $0 \leq i \leq k-1$, $\mathbf{E}[X_A^i] = \sum_{j=1}^k p_j \cdot j^i$. That is,

$$\begin{pmatrix} 1 \\ \mathbf{E}[X_A] \\ \mathbf{E}[X_A^2] \\ \vdots \\ \mathbf{E}[X_A^{k-1}] \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix} \tag{8}$$

Similarly, it holds that

$$\begin{pmatrix} 1 \\ \mathbf{E}[X_B] \\ \mathbf{E}[X_B^2] \\ \vdots \\ \mathbf{E}[X_B^{k-1}] \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \vdots \\ \mathbf{q}_k \end{pmatrix} \tag{9}$$

By Equations (8) and (9), we know that to prove Equation (7), it suffices to show that

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \vdots \\ \mathbf{q}_k \end{pmatrix} = d \cdot \begin{pmatrix} 1/d & 1/d & \cdots & 1/d \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix}. \tag{10}$$

Recall that by Lemma 10, it holds that

$$\begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \vdots \\ \mathbf{q}_k \end{pmatrix} = d \cdot \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix} + (d-1) \cdot \begin{pmatrix} (-1)^1 \binom{k}{1} \\ (-1)^2 \binom{k}{2} \\ (-1)^3 \binom{k}{3} \\ \vdots \\ (-1)^k \binom{k}{k} \end{pmatrix}$$

Substituting $\mathbf{q}_i$ from the above equation to the left hand side of equation (10) gives us that

$$d \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix} + (d-1) \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \cdot \begin{pmatrix} (-1)^1 \binom{k}{1} \\ (-1)^2 \binom{k}{2} \\ (-1)^3 \binom{k}{3} \\ \vdots \\ (-1)^k \binom{k}{i} \end{pmatrix}$$

$$= d \cdot \begin{pmatrix} 1 \\ \mathbf{E}[X_A] \\ \mathbf{E}[X_A^2] \\ \vdots \\ \mathbf{E}[X_A^{k-1}] \end{pmatrix} + (d-1) \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \cdot \begin{pmatrix} (-1)^1 \binom{k}{1} \\ (-1)^2 \binom{k}{2} \\ (-1)^3 \binom{k}{3} \\ \vdots \\ (-1)^k \binom{k}{i} \end{pmatrix} = d \cdot \begin{pmatrix} 1/d \\ \mathbf{E}[X_A] \\ \mathbf{E}[X_A^2] \\ \vdots \\ \mathbf{E}[X_A^{k-1}] \end{pmatrix},$$

where the last equation follows from Claim 1.

On the other hand, by Equation (8), we know that the right hand side of (10) is,

$$d \cdot \begin{pmatrix} 1/d & 1/d & \cdots & 1/d \\ 1 & 2 & \cdots & k \\ 1 & 2^2 & \cdots & k^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{k-1} & \cdots & k^{k-1} \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \vdots \\ \mathbf{p}_k \end{pmatrix} = d \cdot \begin{pmatrix} 1/d \\ \mathbf{E}[X_A] \\ \mathbf{E}[X_A^2] \\ \vdots \\ \mathbf{E}[X_A^{k-1}] \end{pmatrix}.$$

Therefore, Equation (10) holds and thus X_A and X_B have $k-1$ proportional moments. This finishes the proof of the Lemma. ◀

Now we construct class $\mathcal{C}_A$ as follows: $\mathcal{C}_A$ is a class of sequences, and the frequency variable X_A of every sequence A is $\Pr[X_A = i] = p_i$. That is, for every sequence A, the fraction of elements that occur i times is exactly p_i. We can construct $\mathcal{C}_B$ analogously by substituting p_i with q_i.

By construction, sequence A is k-occurrence-free. Consider the sequence B. Suppose that there are l distinct values in B, then at least $q_k \cdot l$ values occur k times in B, which means that B is at least $\frac{q_k \cdot l}{n}$-far from k-occurrence-free. As every value in B occurs in at most k positions, there are at least $\frac{n}{k}$ distinct values, i.e., $l \geq \frac{n}{k}$. Thus, B is at least $\frac{q_k}{k}$-far from k-occurrence-free. According to previous analysis, A and B have $k-1$ proportional moments. The theorem then follows from Lemma 8. ◀

3 A Lower Bound for Testing Subgraph-Freeness

In this section, we give the proof of the lower bound on the query complexity for testing subgraph-freeness, i.e., prove Theorem 2.

Proof of Theorem 2. We give a reduction from the problem of testing k-occurrence of a sequence to the problem of testing H-freeness in the unidirectional model. That is, given an instance of the former problem, i.e., a length-n sequence S such that each element is promised to occur at most k times, we will construct an instance of the H-freeness testing problem, i.e., a directed graph G with $n' = \Theta(n)$ vertices and bounded degree. Then we show that this construction preserves the distances of the properties and any algorithm $\mathcal{A}'$ for testing H-freeness in the unidirectional model can be invoked on G to test if S is

k-occurrence-freeness. In particular, if $\mathcal{A}'$ has query complexity $o(n'^{1-\frac{1}{k}})$, then this implies an algorithm for testing k-occurrence-freeness with query complexity $o(n^{1-\frac{1}{k}})$, contradicting to Theorem 3.

Preprocessing the subgraph *H*. Since H has k source components, we denote these components by $\{C_1, \cdots, C_k\}$. Note that each C_i is a subgraph of H. We use N_{comp} to denote the maximum number of vertices in $\{C_1, C_2, \cdots, C_k\}$, i.e., $N_{\text{comp}} = \max_{i=1,\cdots,k} |V(C_i)|$ where $V(C)$ denotes the vertex set of the graph C. We use C_0 to denote the subgraph induced by the remainder of vertices in $V(H)$ and we call C_0 the center component of H. Let $N_{\text{center}} = |V(C_0)| = |V(H)| - \sum_{i=1}^{k} |V(C_i)|$. Note that since $C_1, \cdots, C_k$ are source components, by definition, no edge exists between different such components. All the edges leaving C_i (for $i = 1, \cdots, k$) are entering C_0. We can first decompose H into source components and the center component (e.g., by using Tarjan's algorithm [21]), index them, and identify all the edges crossing different components in constant time (as the size of H is constant).

We illustrate such a decomposition of a subgraph $\tilde{H}$ in Figure 1. Note that $\tilde{H}$ has 3 source components and 1 center component (see Figure 2). It can be partitioned into four parts such $V(\tilde{C_0}) = \{v_2, v_7\}$, $V(\tilde{C_1}) = \{v_1\}$, $V(\tilde{C_2}) = \{v_3, v_4, v_5\}$, $V(\tilde{C_3}) = \{v_6\}$ as follows. In this example, $N_{\text{comp}} = 3$, $N_{\text{center}} = 2$.

In the construction of the graph G, we will treat each component C_i, $1 \le i \le k$, as a subgraph with N_{comp} vertices. That is, for each such C_i, we add $(N_{\text{comp}} - |V(C_i)|)$ isolated vertices to C_i to obtain a new component C_i' so that $|V(C_i')| = N_{\text{comp}}$. We can reassemble these new components $\{C_1', C_2', \cdots, C_k'\}$ with C_0 to obtain a graph H'.

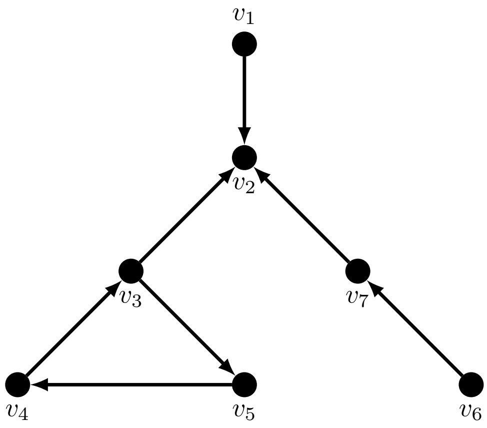
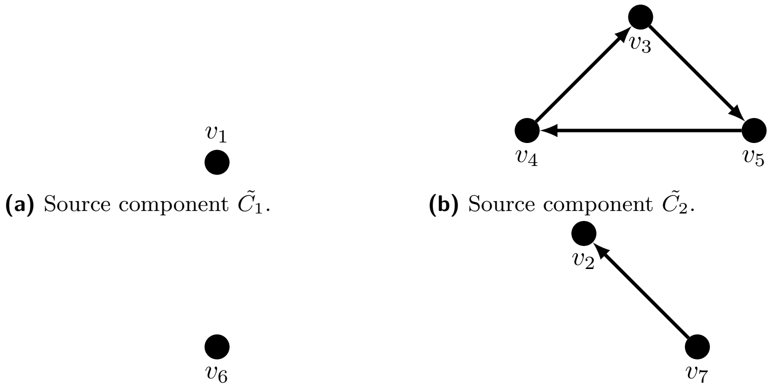


Figure 1 A subgraph $\tilde{H}$.

Now we index each vertex of H' by some integer in $\{1, \cdots, N_{\text{center}} + k \cdot N_{\text{comp}}\}$ as follows. The index set of $V(C_0')$ is $[1, N_{\text{center}}]$, and the index set of $V(C_i')$ is $[N_{\text{center}} + (i-1) \cdot N_{\text{comp}} + 1, N_{\text{center}} + i \cdot N_{\text{comp}}]$, for each $1 \le i \le k$. Furthermore, for each component C_i' with $0 \le i \le k$, we sequentially index the vertices using the corresponding index set according to the lexicographical ordering of the vertices in the aforementioned component decomposition.

Now we describe the reduction. Given a length-n sequence S, the directed graph $G = (V, E)$ can be constructed as follows. We first add n disjoint copies of the subgraph C_0 to G. Then we will add n copies of source components and add some edges from source components to some copy of C_0 constructed before. That is, each element in the sequence corresponds to a source component. Note that there are no edges between different copies of source components. The offline construction is formally described as follows.



(a) Source component $\tilde{C}_1$.

(b) Source component $\tilde{C}_2$.

(c) Source component $\tilde{C}_3$.

(d) Center component $\tilde{C}_0$.

Figure 2 Decomposing $\tilde{H}$ into 3 source components and 1 center component.

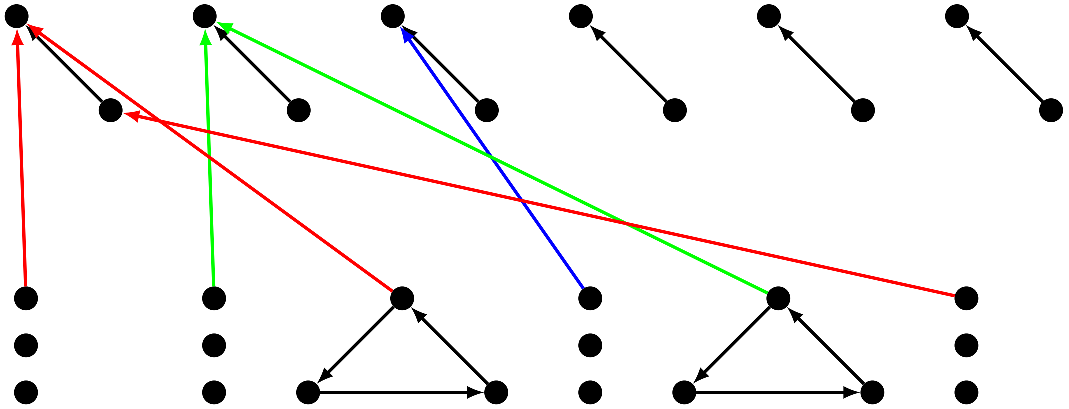


Figure 3 Constructing $\tilde{G}$ from the sequence $\tilde{S}$ and the decomposition of $\tilde{H}$.

Vertex set and vertex indices. We index vertices in G from 1 to $n \cdot (N_{\text{center}} + N_{\text{comp}})$. The vertex set is decomposed into two parts: the *center part* and the *source part*. More precisely, the source part contains n potential source components with vertex indices from 1 to $n \cdot N_{\text{comp}}$, and the center part contains n disjoint copies of the center component C_0 with vertex indices from $n \cdot N_{\text{comp}} + 1$ to $n \cdot (N_{\text{comp}} + N_{\text{center}})$. Furthermore, the vertices in the i-th copy of the source component are indexed from $(i-1) \cdot N_{\text{comp}} + 1$ to $i \cdot N_{\text{comp}}$, while the vertices of the j-th copy of the center component are indexed from $n \cdot N_{\text{comp}} + (j-1) \cdot N_{\text{center}} + 1$ to $n \cdot N_{\text{comp}} + j \cdot N_{\text{center}}$.

Adding components and edges. Add n disjoint copies of C_0 to G. Initialize a size-n array T such that $T_a = 0$ for each $1 \leq a \leq n$. For each $a = 1, 2, \cdots, n$:

1. let b be the value (or element) of S at position a
2. If b is a new value that algorithm sees for the first time, define an array $R_b = \{1, 2, \cdots, k\}$.
3. Uniformly sample a number t from R_b. Add an copy of C'_t. Ignoring isolated vertices in C'_t, add edges between this copy of C'_t and the b-th copy of C_0 in the same way as the connections between their counterparts in the subgraph H. Delete t from R_b. Set $T_a = t$, i.e., the a-th position of S is mapped to a source component C'_t.

Note that by construction, the graph G is d-bounded, and its maximum (in- or out-) degree the same as the maximum (in- or out-) degree of H.

We give an illustration of the above construction in Figure 3. Given a sequence $\tilde{S} = \{1, 2, 1, 3, 2, 1\}$, and a subgraph $\tilde{H}$ as shown in Figure 1. The graph $\tilde{G}$ from the above reduction is shown Figure 3. In this figure, edges of the same color correspond to positions of the same value (or element) in $\tilde{S}$. For example, the 3 red edges correspond to the 3 occurrences of value 1. Together with the corresponding source and center components, these red edges form an copy of $\tilde{H}$.

Construction on the fly. We show that the above construction of G can be done on the fly and each query to G can be answered by querying at most 1 position in S. More precisely, let $\mathcal{A}'$ be an algorithm for testing H-freeness. When $\mathcal{A}'$ queries the i-th outgoing neighbor of a vertex v, we consider the following cases.

If $v > n \cdot N_{\text{comp}}$, then v belongs to a copy of C_0, then we do not need to query sequence S, and we can simply locate the vertex $v' = (v - n \cdot N_{\text{comp}}) \bmod N_{\text{center}}$ in C_0. And by our index in H', we know the corresponding vertex index in H' is also v'. Then we can check the i-th neighbor of v' in H', denoted by v''. Thus we just return $v - v' + v''$.

If $1 \leq v \leq n \cdot N_{\text{comp}}$, then v belongs to a copy of some source component. Calculate $a = \lceil v/N_{\text{comp}} \rceil$ and query the a-th position of S. Let b denote the query answer. If $T_a = 0$, which means that this element is queried for the first time, uniformly sample a type t from the rest of types R_b for value b, and update $T_a = t$; otherwise simply set $t = T_a$. Note that R and T are maintained as described in the construction. Then calculate $v' = v \bmod N_{\text{comp}}$. Now we know that the queried vertex v corresponds to the v'-th vertex in a C'_t component, which is adjacent to the b-th copy of C_0. We can look up vertex $N_{\text{center}} + (t-1) \cdot N_{\text{comp}} + v'$ in H', which is isomorphic to vertex v in G. We use v'' to denote the i-th neighbor of $N_{\text{center}} + (t-1) \cdot N_{\text{comp}} + v'$ in H'. If v'' belongs to the C'_t part in H, we just return $v - v' + v''$. Otherwise, if v'' belongs to a C_0 part, we return $n \cdot N_{\text{comp}} + (b-1) \cdot N_{\text{center}} + v''$.

Thus, any query for a vertex v with $v > n \cdot N_{\text{comp}}$ can be answered without querying S; query for a vertex v with $1 \leq v \leq n \cdot N_{\text{comp}}$ can be answered by making one query to S.

Note that our construction generates a graph G from a distribution $\mathcal{D} = \{G_1, G_2, \cdots\}$. We will show that if S is k-occurrence-free, then any graph from $\mathcal{D}$ is H-free; if S is far from being k-occurrence-free, then every graph in $\mathcal{D}$ is far from H-freeness.

Preserving the distances. Note that in the above construction, if there exists some value occurring k times in S, then these k occurrences of the same value results in k different source components covering $\{C_1, C_2, \cdots, C_k\}$, and they are adjacent to the same center. That is, each element occurring k times in the sequence result in an occurrence of H in G. For each element occurring less than k times, the center corresponding to this value will be adjacent to less than k source components, which in turn implies that H does not occur in this case. We mention that the auxiliary isolated vertices also do not contribute to any occurrence of H.

Thus, if S is k-occurrence-free, then there can not be any occurrence of H, and thus G must be H-free. If S is ε-far from being k-occurrence-free, then there will be at least εn occurrences of H in G. This implies that G is at least ε'-far from H-freeness, for $\varepsilon' = \frac{\varepsilon n}{d(N_{\text{center}} + N_{\text{comp}})n} = \frac{\varepsilon}{d(N_{\text{center}} + N_{\text{comp}})}$.

Putting things together. Let $\mathcal{A}'$ be an algorithm for testing H-freeness with proximity parameter $\varepsilon = \Theta_{k,d}(1)$. Suppose that the query complexity is $o(n'^{1-\frac{1}{k}})$ on an n'-vertex digraph. Now we invoke the algorithm $\mathcal{A}'$ on the graph G that was constructed as before.

As we have seen, each query in G can be answered by making at most 1 query to the sequence S. Furthermore, if S is k-occurrence-free, then G is H-free and if S is ε-far from being k-occurrence-free, then G is ε'-far from H-free, for $\varepsilon' = \frac{\varepsilon}{d\cdot(N_{\text{center}}+N_{\text{comp}})} = \Theta_{k,d}(1)$. Thus, the algorithm $\mathcal{A}'$, together with the construction, also solves the problem of testing k-occurrence-freeness with $o(n^{1-\frac{1}{k}})$ queries, which contradicts Theorem 3. Thus, the query complexity of $\mathcal{A}'$ is $\Omega(n^{1-\frac{1}{k}})$. This finishes the proof of the theorem. ◀

References

1 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009.

2 Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer and System Sciences*, 69(3):354–382, 2004.

3 Michael A Bender and Dana Ron. Testing properties of directed graphs: acyclicity and connectivity. *Random Structures & Algorithms*, 20(2):184–205, 2002.

4 Xi Chen, Tim Randolph, Rocco A Servedio, and Timothy Sun. A lower bound on cycle-finding in sparse digraphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2936–2952. SIAM, 2020.

5 Artur Czumaj, Pan Peng, and Christian Sohler. Relating two property testing models for bounded degree directed graphs. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1033–1045, 2016.

6 Marcel Dall'Agnol, Tom Gur, and Oded Lachish. A structural theorem for local algorithms with applications to coding, testing, and privacy. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1651–1665. SIAM, 2021.

7 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1163–1182. IEEE, 2015.

8 Sebastian Forster, Danupon Nanongkai, Liu Yang, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2046–2065. SIAM, 2020.

9 Oded Goldreich. Introduction to testing graph properties. In *Property testing*, pages 105–141. Springer, 2010.

10 Oded Goldreich. *Introduction to property testing.* Cambridge University Press, 2017.

11 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

12 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

13 Oded Goldreich and Dana Ron. On proximity-oblivious testing. *SIAM Journal on Computing*, 40(2):534–566, 2011.

14 Oded Goldreich and Dana Ron. On sample-based testers. *ACM Transactions on Computation Theory (TOCT)*, 8(2):1–54, 2016.

15 Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM Journal on Computing*, 50(2):788–813, 2021.

16 Frank Hellweg and Christian Sohler. Property testing in sparse directed graphs: strong connectivity and subgraph-freeness. In *European Symposium on Algorithms*, pages 599–610. Springer, 2012.

17 Hiro Ito, Areej Khoury, and Ilan Newman. On the characterization of 1-sided error strongly testable graph properties for bounded-degree graphs. *computational complexity*, 29(1):1–45, 2020.

18 Yaron Orenstein and Dana Ron. Testing eulerianity and connectivity in directed sparse graphs. *Theoretical Computer Science*, 412(45):6390–6408, 2011.

19 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.

20 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

21 Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

22 Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6):1927, 2011.

23 Yuichi Yoshida and Hiro Ito. Testing k-edge-connectivity of digraphs. *Journal of systems science and complexity*, 23(1):91–101, 2010.