A Characterization of Easily Testable Induced Subgraphs*

Noga Alon † Asaf Shapira ‡

Abstract

Let H be a fixed graph on h vertices. We say that a graph G is *induced* H-free if it does not contain any *induced* copy of H. Let G be a graph on n vertices and suppose that at least ϵn^2 edges have to be added to or removed from it in order to make it induced H-free. It was shown in [5] that in this case G contains at least $f(\epsilon, h)n^h$ induced copies of H, where $1/f(\epsilon, h)$ is an extremely fast growing function in $1/\epsilon$, that is independent of n. As a consequence, it follows that for every H, testing induced H-freeness with one-sided error has query complexity independent of n. A natural question, raised by the first author in [1], is to decide for which graphs H the function $1/f(\epsilon, H)$ can be bounded from above by a polynomial in $1/\epsilon$. An equivalent question is for which graphs H, can one design a one-sided error property tester for testing induced H-freeness, whose query complexity is polynomial in $1/\epsilon$. We settle this question almost completely by showing that, quite surprisingly, for any graph other than the paths of lengths 1,2 and 3, the cycle of length 4, and their complements, no such property tester exists. We further show that a similar result also applies to the case of directed graphs, thus answering a question raised by the authors in [9]. We finally show that the same results hold even in the case of two-sided error property testers. The proofs combine combinatorial, graph theoretic and probabilistic arguments with results from additive number theory.

1 Preliminaries

1.1 Definitions

All graphs and directed graphs (=digraphs) considered here are finite and have no loops and no parallel edges. Let P be a property of graphs, that is, a family of graphs closed under isomorphism. A graph G with n vertices is *ϵ-far from satisfying* P if no graph $\tilde{G}$ with the same vertex set, which differs from G in at most ϵn^2 places, (i.e., can be constructed from G by adding and removing at most ϵn^2 edges), satisfies P. An *ϵ-tester*, or *property tester*, for P is a randomized algorithm which,

*A preliminary version of this paper appeared in the Proc. of the 15^{th} Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM Press (2004), pp. 935-944.

†Schools of Mathematics and Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: noga@math.tau.ac.il. Research supported in part by a USA-Israeli BSF grant, by the Israel Science Foundation and by the Hermann Minkowski Minerva Center for Geometry at Tel Aviv University.

‡School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: asafico@math.tau.ac.il. This work forms part of the author's Ph.D. Thesis.

given the quantity n and the ability to make queries whether a desired pair of vertices of an input graph G with n vertices are adjacent or not, distinguishes with probability at least $\frac{2}{3}$ between the case of G satisfying P and the case of G being ϵ-far from satisfying P. Such an ϵ-tester is a *one-sided* ϵ-tester if when G satisfies P the ϵ-tester determines that this is the case (with probability 1). The ϵ-tester is a *two-sided* ϵ-tester if it may determine that G does not satisfy P even if G satisfies it. Obviously, the probability $\frac{2}{3}$ appearing above can be replaced by any constant smaller than 1, by repeating the algorithm an appropriate number of times.

The property P is called *strongly-testable*, if for every fixed $\epsilon > 0$ there exists a one-sided ϵ-tester for P whose total number of queries is bounded only by a function of ϵ, which is independent of the size of the input graph. This means that the running time of the algorithm is also bounded by a function of ϵ only, and is independent of the input size.

In what follows we denote by P_2, P_3 and P_4 the paths of lengths 1,2 and 3 (which have 2,3 and 4 vertices, respectively), and by C_4, the cycle of length 4. We measure query-complexity by the number of vertices sampled, assuming we always examine all edges spanned by them. For a fixed graph H, let P^*_H denote the property of being induced H-free. Therefore, G satisfies P^*_H if and only if it contains no induced subgraph isomorphic to H. We define P_H to be the property of being (not necessarily induced) H-free. Therefore, G satisfies P_H if and only if it contains no copy of H. Thus, for example, for $H = C_4$, any clique of size at least 4 satisfies P^*_H but does not satisfy P_H.

1.2 Background

The general notion of property testing was first formulated by Rubinfeld and Sudan [27], who were motivated mainly by its connection to the study of program checking. The study of the notion of testability for combinatorial objects, and mainly for labelled graphs, was introduced by Goldreich, Goldwasser and Ron [22], who showed that all graph properties describable by the existence of a partition of a certain type, and among them k-colorability, have efficient ϵ-testers. The notion of property testing has been investigated in other contexts as well, including the context of regular languages, [6], functions [20], [8], [3], [18], [19], computational geometry [15], [4], graph and hypergraph coloring [7], [14], [8], [11] and other contexts. See [21], [26] and [17] for surveys on the topic.

2 The Main Results

2.1 Related Work

In [5] it is shown that every first order graph property without a quantifier alternation of type "$\forall\exists$" has ϵ-testers whose query complexity is independent of the size of the input graph. It follows from the main result of [5] that for every fixed H, the property P^*_H is strongly testable. Though the query complexity sufficient for testing P^*_H is independent of n, it has a huge dependency on $1/\epsilon$ (the fourth function in the Ackerman Hierarchy, which is a tower of towers of exponents). In [2], it was shown, using Szemerédi's Regularity Lemma [28], that for every fixed H, the property P_H is also strongly testable. This result was generalized to the case of directed graphs (=digraphs) in [9], by first proving a directed version of the regularity lemma. In the above two cases the query complexity is also huge, a tower of 2's of height polynomial in $1/\epsilon$. For some graphs, however, there are obviously much more efficient property testers than the ones guaranteed by the above general results. For example for the case of H being an edge, there is obviously a one-sided error property tester for both P_H and P^*_H, whose query complexity is $\Theta(1/\epsilon)$. A natural question is therefore, to decide for which graphs H can one design a one-sided error property tester for P_H or P^*_H, whose query complexity would be bounded by a polynomial of $1/\epsilon$. We call a property P *easily testable* if there is a one-sided error property tester for P whose query complexity is polynomial in $1/\epsilon$. If no such property tester exists we say that P is *hard to test*.

In [1] it is shown that for an *undirected* graph H, P_H is easily testable if and only if H is bipartite. The authors of [9] obtain a precise characterization of all the *directed* graphs H for which P_H is easily testable. As is evident from this characterization, recognizing these digraphs is rather difficult. Indeed, it is shown in [9] that deciding whether for a digraph H, P_H is easily testable, is NP-complete. The next natural steps, suggested in [1] and [9], are therefore to give characterizations of the graphs and digraphs H for which P^*_H is easily testable. In this paper we give nearly complete such characterizations in both cases.

2.2 The new results

Our first new result here is the following:

Theorem 1 *Let H be a fixed undirected graph other than P_2, P_3, P_4, C_4 and their complements. Then, there exists a constant $c = c(H) > 0$ such that the query-complexity of any one-sided error ϵ-tester for P^*_H is at least*

$$\left(\frac{1}{\epsilon}\right)^{c\log(1/\epsilon)}.$$

As P_2-freeness can obviously be tested with query complexity $\Theta(1/\epsilon)$, the following theorem, together with the above theorem, supplies a complete characterization for the graphs H for which P^*_H is easily testable, except for the case of P_4, C_4 and its complement (the complement of P_4 is also P_4).

Theorem 2 *There is a one-sided error property tester for testing P_3-freeness, with query complexity*

$$O(\log(1/\epsilon)/\epsilon).$$

We also prove the following theorem, which is analogous to Theorem 1, only with respect to digraphs.

Theorem 3 *Let H be a fixed digraph on at least 5 vertices. Then, there exists a constant $c = c(H) > 0$ such that the query-complexity of any one-sided error ϵ-tester for P^*_H is at least*

$$\left(\frac{1}{\epsilon}\right)^{c\log(1/\epsilon)} .$$

We can actually prove a super-polynomial (in $1/\epsilon$) lower bound for the query complexity of P^*_H for some of the digraphs H on at most 4 vertices as well, see subsection 4.1.

We finally show that Theorems 1 and 3 can also be extended to the cases of two-sided error property testers.

Theorem 4 *The lower bounds of Theorems 1 and 3 hold for two-sided error property testers as well.*

An interesting consequence of the above results is that the class of graphs for which P^*_H is easily testable, is nearly trivial (as it contains graphs on at most 4 vertices), however, it is provably not totally trivial, as $P^*_{P_3}$ is easily testable. Note also the sharp dichotomy between the efficient one-sided error property-testers for $P^*_{P_2}$ and $P^*_{P_3}$, and the fact that for almost all the other graphs H, the property P^*_H has no property tester with query complexity polynomial in $1/\epsilon$ even if one is willing to settle for *two-sided* error.

2.3 Organization

The proof of Theorem 2 appears in Section 3. The lower bound proved by Theorem 1 is established in section 4. To prove this result we have to construct, for any graph H (other than the ones mentioned in the theorem) and any small $\epsilon > 0$, a graph G which is ϵ-far from being induced H-free and yet contains relatively few induced copies of H. The proof of this part, described in Section 4, uses the approaches of [1] and [9], but requires several additional ideas. It applies certain constructions in additive number theory, based on (simple variants of) the construction of Behrend [10] of dense subsets of the first n integers without three-term arithmetic progressions. The proof of Theorem 3 also appears in section 4. In Section 5 we give the proof of Theorem 4 which extends the lower bounds of Theorems 1 and 3 to the more general cases of two-sided error property-testers. The final section, Section 6, contains some concluding remarks and open problems.

Throughout this paper we assume, whenever this is needed, that the number of vertices n of the graph or digraph G considered is sufficiently large, and that the error parameter ϵ is sufficiently small. In order to simplify the presentation, we omit all floor and ceiling signs whenever these are not crucial, and make no attempt to optimize the absolute constants. When we later refer to a graph H as being easy/hard to test, we mean that P^*_H is easy/hard to test.

3 An Easily Testable Graph

In this section we describe the proof of Theorem 2. For ease of notation, we denote by P the property $P^*_{P_3}$, that is, being induced P_3-free. The property-tester for P works as follows: it picks a random subset of, say, $10\log(1/\epsilon)/\epsilon$ vertices, and checks if there is an induced copy of P_3 spanned by the set. It declares G to be induced P_3-free if and only if it finds no induced copy of P_3. If G satisfies P, the algorithm clearly always answers correctly. We therefore only have to show that if G is ϵ-far from satisfying P, the algorithm finds an induced copy of P_3 with probability at least $2/3$.

Let *High* denote the set of vertices of such a graph G whose degree is at least $\frac{\epsilon}{4}n$. Note that intuitively, the vertices that belong to $High$ have the highest contribution to G being ϵ-far from satisfying P. We formulate this intuition as follows:

Claim 3.1 *Let $W \subseteq V(G)$ contain all but at most $\frac{\epsilon}{4}n$ of the vertices of* High*. Then the induced subgraph of G on W is at least $\frac{\epsilon}{2}$-far from satisfying P.*

Proof: Assume this is not the case. Then we can make less than $\frac{\epsilon}{2}n^2$ changes within W and get a graph that contains no induced copy of P_3 within W. We then remove all the edges that touch a vertex not in $High \cup W$ (as these vertices do not belong to *High*, there are at most $n \cdot \frac{\epsilon}{4}n$ such edges), and any edge that touches a vertex in $High \setminus W$ (there are at most $\frac{\epsilon}{4}n \cdot n$ such edges as by the assumption $|High \setminus W| \leq \frac{\epsilon}{4}n$). We thus get a graph that satisfies P. As altogether we make less than ϵn^2 changes in G, this contradicts the assumption that G is ϵ-far from satisfying P. ■

We call a set $A \subseteq V(G)$ *Good* if all but at most $\frac{\epsilon}{4}n$ of the vertices that belong to *High* have a neighbor in A.

Claim 3.2 *A randomly chosen subset $A \subseteq V(G)$ of size $8\log(1/\epsilon)/\epsilon$ is* Good *with probability at least 7/8.*

Proof: Let A be a randomly chosen subset of size $8\log(1/\epsilon)/\epsilon$, and consider a vertex $v \in High$. As v has at least $\frac{\epsilon}{4}n$ neighbors, the probability that A does not contain any neighbor of v is at most

$$\left(1 - \frac{\epsilon}{4}\right)^{8\log(1/\epsilon)/\epsilon} \leq \epsilon^2 \leq \epsilon/32,$$

where we assumed that $\epsilon < 1/32$. As $High$ is of size at most n, we conclude that the expected number of vertices that belong to $High$ and have no neighbor in A, is at most $\frac{\epsilon}{32}n$. By Markov's inequality, with probability at least $7/8$, the number of these vertices is at most $\frac{\epsilon}{4}n$. ■

We will use the following simple and well known observation about the structure of induced P_3-free graphs: A graph is induced P_3-free if and only if it is the disjoint union of cliques.

Proof of Theorem 2 We first choose a random subset A of size $8\log(1/\epsilon)/\epsilon$, and assume that it is *Good*. If A contains an induced copy of P_3 we are done. Otherwise, let W be the set of all the vertices $v \in V \setminus A$ that have at least one neighbor in A. As G is by assumption ϵ-far from satisfying P, and A is by assumption *Good*, we get from Claim 3.1 that the induced subgraph on W is at least $\frac{\epsilon}{2}$-far from satisfying P.

As we assumed that A contains no induced copy of P_3, we get that there is a unique partition of A into cliques $C_1, \ldots, C_r$. If a vertex $v \in W$ is connected to $u \in C_i \subseteq A$, it follows that if W can be partitioned into cliques $D_1, \ldots, D_k$, where for $1 \leq i \leq r$, $C_i \subseteq D_i$, then v would have to belong to D_i. For each vertex $v \in W$ that is connected to $u \in C_i \subseteq A$, assign v the number i. If u is connected to vertices in A that belong to different C_i, then pick any of these numbers. This numbering induces a partition of all the vertices of W into r subsets. As W is at least $\frac{\epsilon}{2}$-far from satisfying P, there are at least $\frac{\epsilon}{2}n^2$ pairs of vertices $u, v \in W$, such that either u and v should belong to the same D_i, but u and v are not connected, or u and v should belong to different subsets D_i, yet u and v are connected. Therefore, choosing a set B of $8/\epsilon$ randomly chosen pairs of vertices fails to find such a violating pair with probability at most $(1-\epsilon/2)^{8/\epsilon} \leq \frac{1}{8}$. By Claim 3.2, the probability of A failing to be *Good* is at most $\frac{1}{8}$, and the probability of B not containing any of the required pairs of vertices is also at most $\frac{1}{8}$. Hence, with probability at least $\frac{3}{4}$ the induced subgraph on $A \cup B$ is not induced P_3-free. As $|A| + |B| = O(\log(1/\epsilon)/\epsilon)$ the proof is complete. ■

4 Hard to Test Graphs and Digraphs

In this section we give the proofs of Theorems 1 and 3. The approach uses a construction in additive number theory, which uses the technique of Behrend [10], used to construct dense sets of integers with no three-term arithmetic progressions. A set $X \subseteq M = \{1, 2, \ldots, m\}$ is called *h-sum-free* if for every pair of positive integers $a, b \leq h$, if $x, y, z \in X$ satisfy the equation $ax + by = (a+b)z$ then $x = y = z$. That is, whenever $a, b \leq h$, the only solution to the equation that uses values from X, is one of the $|X|$ trivial solutions. We need the following lemma (a similar one appears in [16]):

Lemma 4.1 *For every positive integer m, there exists an h-sum-free subset $X \subset M = \{1, 2, \ldots, m\}$ of size at least*

$$|X| \geq \frac{m}{e^{10\sqrt{\log h \log m}}} \tag{1}$$

Proof: Let d and r be integers (to be chosen later) and define

$$S_r = \left\{ \sum_{i=0}^{k} x_i d^i \mid x_i < \frac{d}{2h} \ (0 \leq i \leq k) \ \wedge \sum_{i=0}^{k} x_i^2 = B \right\},$$

where $k = \lfloor \log m / \log d \rfloor - 1$. For the rest of the proof, the best way to view the numbers $x \in S_r$ is as represented in base d, where $x_k, \ldots, x_0$ are the "digits" of x. Also, note that be the choice of k, for any r we have $S_r \subseteq [n]$.

We claim that for every d and r, S_r is h-sum-free. Assume to the contrary that there are $x, y, z \in S_r$ that satisfy the equation $ax + by = (a+b)z$, where $a, b \leq h$ are positive integers and

$$x = \sum_{i=0}^{k} x_i d^i, \quad y = \sum_{i=0}^{k} y_i d^i, \quad z = \sum_{i=0}^{k} z_i d^i.$$

As by definition $x_i, y_i, z_i < d/2h$, and $a, b \le h$ we conclude that there is no carry in the base d addition of the numbers in S_r. In other words, we have for every $0 \le i \le k$

$$ax_i + by_i = (a+b)z_i.$$

This means that z_i is a weighted average of x_i and y_i. Combined with the fact that the function $f(z) = z^2$ is convex, Jensen's inequality implies that

$$ax_i^2 + by_i^2 \ge (a+b)z_i^2,$$

and that the inequality is strict unless all three numbers x_i, y_i and z_i are equal. However, if for some i the inequality is strict, we have

$$a\sum_{i=0}^{k} x_i^2 + b\sum_{i=0}^{k} y_i^2 > (a+b)\sum_{i=0}^{k} z_i^2$$

which is impossible as by definition of S_r

$$\sum_{i=0}^{k} x_i^2 = \sum_{i=0}^{k} y_i^2 = \sum_{i=0}^{k} z_i^2 = r.$$

Thus, $x_i = y_i = z_i$ for all i, and S_r is indeed h-sum-free.

We complete the proof by showing that for some r, the set S_r is of the required size in (1). As the "digits" in any set S_r are bounded by $d/2h$, the integer r in the definition of S_r satisfies $r \le (k+1)(d/2h)^2 < kd^2$. For the same reason, the union of the sets S_r has size $(d/2h)^{k+1} > (d/2h)^k$. It follows that for some r, the set S_r satisfies $|S_r| > (d/2h)^k/kd^2$. Setting $d = e^{\sqrt{\log m \log h}}$ (and therefore $k \approx \sqrt{\log m/\log h}$), we obtain (1) as needed. ■

We proceed with the proofs of Theorems 1 and 3. It is convenient to start the discussion with digraphs and then obtain the results for undirected graphs as a special case, (as they can be viewed as symmetric digraphs).

An *s-blow-up* of a digraph $H = (V(H), E(H))$ on h vertices is the digraph obtained from H by replacing each vertex $v_i \in V(H)$ by an independent set I_i of size s, and each directed edge $(v_i, v_j) \in E(H)$, by a complete bipartite directed subgraph whose vertex classes are I_i and I_j, and whose edges are directed from I_i to I_j. Note that if we take an s-blow-up of H, we get a digraph on sh vertices that contains s^h *induced* copies of H, where each vertex of the copy belongs to a different blow-up of a vertex from H (simply pick one vertex from each independent set). We call these induced copies the *special copies* of the blow-up. As each pair of vertices in the blow-up is contained in at most s^{h-2} special copies of H, it follows that adding or removing an edge from the graph can destroy at most s^{h-2} special copies of H. We conclude that one must add or remove at least $s^h/s^{h-2} = s^2$ edges from the blow-up in order to destroy all its special copies of H.

For the proofs of Theorems 1 and 3, we will need the following lemma, in which a triangle in a digraph is simply three vertices u, v, w, such that there is at least one edge between each of the three pairs.

Lemma 4.2 *For every fixed digraph H on h vertices, that contains at least one triangle, there is a constant $c = c(H) > 0$, such that for every positive $\epsilon < \epsilon_0(H)$ and every integer $n > n_0(\epsilon)$, there is a digraph G on n vertices which is ϵ-far from being induced H-free, and yet contains at most $\epsilon^{c \log(1/\epsilon)} n^h$ induced copies of H.*

Proof: Given a small $\epsilon > 0$, let m be the largest integer satisfying

$$\frac{1}{h^4 e^{10\sqrt{\log m \log h}}} \geq \epsilon. \tag{2}$$

It is easy to check that this m satisfies

$$m \geq \left(\frac{1}{\epsilon}\right)^{c \log(1/\epsilon)}, \tag{3}$$

for an appropriate $c = c(h) > 0$. Let $X \subset \{1, 2, \ldots, m\}$ be as in Lemma 4.1. Call the vertices of H $v_1, \ldots, v_h$, and let $V_1, V_2, \ldots V_h$ be pairwise disjoint sets of vertices, where $|V_i| = im$ and we denote the vertices of V_i by $\{1, 2, \ldots, im\}$, where, with a slight abuse of notation, we think on the sets V_i as being pairwise disjoint. We now construct a graph F whose vertex set is the union of the sets $V_1, \ldots, V_h$. For each j, $1 \leq j \leq m$, for each $x \in X$ and for each directed edge (v_p, v_q) of H, let $j + (p-1)x \in V_p$ have an outgoing edge pointed to $j + (q-1)x \in V_q$. In other words, for each $1 \leq j \leq m$ and $x \in X$, the graph F contains a copy of H, which is spanned by the vertices $j, j + x, j + 2x, \ldots, j + (h-1)x$. Note that each of these $m|X|$ copies of H is spanned by a set of vertices that forms an arithmetic progression whose first element is j and whose difference is x. A crucial implication is that F contains $m|X|$ copies of H, such that each pair of copies have at most one common vertex. As each edge of F belongs to one of these copies, these $m|X|$ copies of H in F are in particular *induced*. In what follows we call these $m|X|$ induced copies of H in F, the *essential* copies of H in F. Finally, define

$$s = \left\lfloor \frac{n}{|V(F)|} \right\rfloor = \left\lfloor \frac{2n}{h(h+1)m} \right\rfloor$$

and let G be the s-blow-up of F (together with some isolated vertices, if needed, to make sure that the number of vertices is precisely n). Claims 4.1 and 4.2 below complete the proof of this lemma. ■

Claim 4.1 *The digraph G defined in the proof of Lemma 4.2 is ϵ-far from being induced H-free.*

Proof: The main idea of the proof is to show that adding or removing an edge from G can destroy special copies of H that belong to *at most* one of the blow-ups of the essential copies of H in F. To this end, consider two essential copies of H in F, H_1 and H_2. As was noted above, H_1 and H_2 are induced copies of H in F, which share at most one vertex in F. It follows that their corresponding blow-ups in G, denoted by T_1 and T_2, will share at most one common independent set. As T_1 and T_2 share at most one common independent set, a special copy of H in T_1 and a special copy of H in T_2 share at most one common vertex (recall that a special copy in a blow-up of H has precisely one vertex in each of the independent sets). We conclude that adding or removing an edge from G, can either destroy special copies of H that belong to T_1, or special copies of H that belong to T_2 (or not destroy any copies at all). As was explained above, in order to destroy all the special copies of an s-blow-up of H, one needs to add or remove at least s^2 edges from the blow-up. As G contains $m|X|$ blow-ups of essential copies of H, and each of these essential copies is induced in F, we conclude that one has to add or delete at least

$$s^2 m|X| = \frac{4n^2 m|X|}{h^2(h+1)^2 m^2} \geq \frac{|X|n^2}{h^4 m} \geq \frac{n^2}{h^4 e^{10\sqrt{\log m \log h}}} \geq \epsilon n^2 \tag{4}$$

edges in order to make G induced H-free. The second inequality follows from the lower bound on $|X|$ guaranteed by Lemma 4.1, and the third from (2). We conclude that G is indeed ϵ-far from being induced H-free. ■

Claim 4.2 *The digraph G defined in the proof of Lemma 4.2 contains at most $\epsilon^{c\log(1/\epsilon)} n^h$ induced copies of H.*

Proof: As H contains at least one triangle, and each triangle belongs to at most $\binom{n}{h-3} \leq n^{h-3}$ copies of H, it is enough to show that G contains at most $\epsilon^{c\log(1/\epsilon)} n^3$ triangles. Consider a partition of the vertices of G into h subsets $U_1, \ldots, U_h$, where U_i contains the im independent sets that resulted from the blow-ups of the im vertices that belonged to V_i in F. Notice that if we show that the induced subgraph of G on any three of the subsets $U_1, \ldots, U_h$ contains at most $\epsilon^{c'\log(1/\epsilon)} n^3$ triangles, then the total number of triangles in G is at most $\binom{h}{3}\epsilon^{c'\log(1/\epsilon)} n^3$, which is still at most $\epsilon^{c\log(1/\epsilon)} n^3$.

Fix any three subsets U_i, U_j, U_k such that $1 \leq i < j < k \leq h$. Recall that G is a blow-up of F, and that we denote by I_v the independent set of vertices in G, which replaced the vertex $v \in V(F)$. As there are no edges within these sets any triangle spanned by them must have exactly one vertex in each set. Note, that if the sets span a triangle whose vertices belong to the independent sets $I_x \subseteq U_i,\ I_y \subseteq U_j,\ I_z \subseteq U_z$, then as G is a blow-up of F, the vertices $x \in V_i,\ y \in V_j,\ z \in V_k$ in F must also span a triangle. Conversely, if $x \in V_i,\ y \in V_j,\ z \in V_k$ span a triangle in F, then for every choice of three vertices $u \in I_x \subseteq U_i,\ v \in I_y \subseteq U_j,\ w \in I_z \subseteq U_k$, the vertices u, v, w span a triangle in G. It follows that the number of triangles spanned by U_i, U_j, U_k is exactly s^3 times the number of triangles spanned by V_i, V_j, V_k.

If the vertices v_i, v_j, v_k, do not span a triangle in H, then by the definition of F, V_i, V_j, V_k do not span a triangle, and so do U_i, U_j, U_k in G, and we are done. If v_i, v_j, v_k span a triangle in H, then by the definition of F for any triangle spanned by V_i, V_j, V_k, there are $x, y \in X$ and $1 \leq t \leq im$, such that the three vertices of this triangle are

$$t \in V_i, \quad t + (j-i)x \in V_j, \quad t + (j-i)x + (k-j)y \in V_k.$$

The reason is that by definition of F, any edge from V_i to V_j connects some integer $t \in V_i$ to another integer $t + (j-i)x \in V_j$, where $x \in X$. The same applies also to edges connecting vertices from V_j to

V_k. As this is a triangle, there must also be an edge connecting $t \in V_i$ to $t+(j-i)x+(k-j)y \in V_k$, hence there is some $z \in X$ such that

$$t + (k-i)z = t + (j-i)x + (k-j)y.$$

We conclude that the following equation in positive coefficients, whose values are at most h (recall $1 \leq i < j < k \leq h$), holds

$$(j-i)x + (k-j)y = (k-i)z.$$

As X is h-sum free, it follows that $x = y = z$. Therefore, V_i, V_j, V_k span precisely $m|X|$ triangles, which are spanned by the vertices

$$t + (i-1)x \in V_i, \qquad t + (j-1)x \in V_j, \qquad t + (k-1)x \in V_k,$$

for every possible choice of $t \in \{1, \ldots, m\}$ and $x \in X$. We conclude that U_i, U_j, U_k span

$$m|X|s^3 < m^2(n/m)^3 \leq n^3/m$$

triangles. As by (3), $m \geq (1/\epsilon)^{c\log(1/\epsilon)}$, the proof is complete. ■

The proofs of Theorems 1 and 3 now follow easily from the above lemma.

Proof of Theorem 1: Let H be a fixed graph on h vertices. A simple yet crucial observation is that for every graph H testing P^*_H is equivalent to testing $P^*_{\overline{H}}$, where $\overline{H}$ is the complement of H. Note, that this relation does not hold for testing P_H. It follows that in order to prove a lower bound for testing P^*_H, we may prove a lower bound for testing $P^*_{\overline{H}}$.

Given a one-sided error ϵ-tester for testing P^*_H we may assume, without loss of generality, that it queries all pairs of a uniformly at random chosen set of vertices (otherwise, as explained in [5], every time the algorithm queries a pair of vertices we make it query also all pairs containing a vertex of the new pair and a vertex from previous queries. See also [23] for a more detailed proof of this statement.) As the algorithm is a one-sided-error algorithm, it can report that G does not satisfy P^*_H only if it finds an induced copy of H in it. Observe, that if the tester picks a random subset of x vertices, and an input graph contains only $\epsilon^{c\log(1/\epsilon)}n^h$ induced copies of H, then the expected number of induced copies of H spanned by x is at most $x^h\epsilon^{c\log(1/\epsilon)}$, which is far smaller than 1 unless x exceeds $(1/\epsilon)^{c'\log(1/\epsilon)}$ for some $c' = c'(H) > 0$. It follows by Markov's inequality that the tester finds an induced copy of H with negligible probability. It is therefore enough to show that for any undirected graph H, other than P_2, P_3, P_4, C_4 and their complements, there is a graph G on n vertices which is ϵ-far from satisfying P^*_H, yet contains only $\epsilon^{c\log(1/\epsilon)}n^h$ induced copies of H. Combined with the first paragraph of this proof, it is enough to show this for either H or $\overline{H}$.

If $h \geq 6$, then it follows from the simplest result in Ramsey Theory (c.f., e.g., [24], page 1) that either H or $\overline{H}$ must contain a triangle. Hence, assuming that H contains a triangle, we can use Lemma 4.2 to construct a graph G on n vertices which is ϵ-far from satisfying P^*_H and yet contains at most $\epsilon^{c\log(1/\epsilon)}n^h$ induced copies of H. For $h = 5$, the only graph H, such that neither H nor $\overline{H}$ contains a triangle is C_5 (the cycle of length 5, whose complement is also C_5). In this case we can use the fact that C_5 is the core of itself to prove that $P^*_{C_5}$ is not easily testable. See subsection 4.1 for more details. As for $h = 2, 3, 4$ the only graphs H for which H and $\overline{H}$ are triangle-free are P_2, P_3, P_4, C_4 and their complements, the proof is complete. ■

Proof of Theorem 3: The proof is similar to the proof of Theorem 1. One only has to note again that for every digraph H, on at least 6 vertices, either H or $\overline{H}$ contains a triangle, and that the only digraph on 5 vertices which does not have this property is the digraph D_5 obtained from C_5, by replacing each undirected edge with two anti-parallel directed edges. We discuss this special case in subsection 4.1. Though the theorem does not explicitly state it, we can also conclude from Lemma 4.2 that the same lower bound applies for any digraph H on 3 or 4 vertices such that either H or $\overline{H}$ contains a triangle. In subsection 4.1 we discuss some more digraphs for which we can obtain similar bounds. ■

4.1 Graphs which are cores of themselves

In this subsection we briefly argue how to use the results of [9] in order to obtain lower bounds for some digraphs on 3,4 and 5 vertices. We first need some definitions. A homomorphism from digraph H to digraph K is a mapping $\varphi : V(H) \mapsto V(K)$ that maps edges of H to edges of K, i.e. $(u, v) \in E(H) \Rightarrow (\varphi(u), \varphi(v)) \in E(K)$. The *core* of a digraph H, is the smallest *subgraph* of H (with respect to number of edges) to which there is a homomorphism from H. In [9] the authors establish a lower bound similar to those of Theorems 1 and 2 for testing P_H for any digraph H whose core contains at least one cycle of length at least 3. As in the proof of Lemma 4.2, the main ingredient of the proof (Lemma 8 in [9]) is a construction of a digraph that is ϵ-far from being H-free and yet contains relatively few copies of H. Though it is not explicitly stated in [9], in case H is the core of itself, the constructed graph is actually also ϵ-far from being *induced* H-free, and contains relatively few *induced* copies of H. Thus we can use the result of [9] to obtain similar lower bounds for any digraph H on 3,4 or 5 vertices such that either H or $\overline{H}$ is the core of itself and contains a cycle of length at least 3. This in particular holds for C_5, and therefore also for D_5, as testing P_{C_5} is a special case of testing P_{D_5}. As was noted in [9], any directed cycle C that contains a non equal number of forward edges and backward edges is the core of itself. Thus, any digraph on 4 vertices that contains such a cycle of length 4 (e.g. a Hamilton cycle) is the core of itself, and we can use the result of [9] to obtain a lower bound for this case as well.

5 Two-Sided Error Property-Testers

For the proof of Theorem 4 we apply Yao's principle [29], by constructing, for every fixed graph H, for which a lower bound was established in Theorems 1 and 3, two distributions D_1 and D_2, where D_1 consists of graphs which are ϵ-far from satisfying P^*_H with probability $1 - o(1)$ (where the $o(1)$

term tends to 0 as ϵ tends to zero), while D_2 consists of graphs which satisfy P^*_H. We then show that any deterministic algorithm, which makes a small number of queries (adaptively) cannot distinguish with non-negligible probability between D_1 and D_2. We prove Theorem 4 for the case of digraphs, as it is clear that the case of undirected graphs follows as a special case. For the case of H being the graph obtained from C_5 by replacing each edge by a cycle of length 2, we can use the fact that this graph is the core of itself (as we did for one sided error in subsection 4.1) to prove that $P^*_{C_5}$ has no two-sided ϵ-tester with query complexity polynomial in $1/\epsilon$. We thus assume that H is a graph on at least 6 vertices. As in the proofs of Theorems 1 and 3, testing P^*_H with two-sided error has the same query complexity as testing $P^*_{\overline{H}}$, thus we assume that H contains at least one triangle.

Proof of Theorem 4: Let H be a fixed digraph which contains at least one triangle. Given n and ϵ, let X, m and the sets V_i be as in the proof of Lemma 4.2. Construct the digraph F just as in the proof of Lemma 4.2, and remember that it consists of $m|X|$ pairwise edge disjoint copies of H which we called the essential copies of H in F (though it may well contain additional copies of H).

To construct D_1 which consists of digraphs that are ϵ-far from satisfying P^*_H with high probability, we first construct F'_1 by removing each of the $m|X|$ essential copies of H, randomly and independently, with probability $1 - 1/|E(H)|$. We then create G_1 by taking an s blow up of F'_1, adding isolated vertices, if needed. Finally, D_1 consists of all randomly permuted copies of such digraphs G_1. It follows from a standard Chernoff bound, that with probability $1 - o(1)$, at least $m|X|/2|E(H)|$ essential copies of H are left in F'_1, where the $o(1)$ term tends to 0, as ϵ tends to 0. Similar to the derivation of (4), it is easy to show that if $m|X|/2|E(H)|$ of these copies of H are left in F'_1, the graph G_1 is ϵ-far from satisfying P^*_H. It follows that with probability $1 - o(1)$, a member of D_1 is ϵ-far from satisfying P^*_H.

The distribution D_2 of digraphs that satisfy P^*_H, is defined by first constructing F'_2 by randomly, independently and uniformly picking from each of the $m|X|$ essential copies of H a single edge, and removing all the other edges of that copy. We then create G_2 by taking an s blow up of F'_2 adding isolated vertices, if needed. Finally, D_2 consists of all randomly permuted copies of such digraphs G_2. The main argument of Lemma 4.2, states that the graph F defined in the lemma contains only triangles whose three edges belong to one of the essential copies of H. Hence, keeping a single edge from each of these copies results in a triangle free graph, and in particular all the graphs in G_2 satisfy P^*_H.

As in the proof of Lemma 4.2, denote by I_v the independent set of vertices in G_1 (or G_2) that replaces the vertex $v \in V(F)$. Now consider a set of vertices S in G_1 (or G_2) and its natural projection to a subset of $V(F)$ (namely, for each vertex $u \in I_v$ we consider the vertex v in F) which we also denote by S with a slight abuse of notation. Suppose S has the property that it does not contain more than two vertices from any one of the essential copies of H.

If this property holds, then each edge spanned by S is contained in a different essential copy of H. Therefore, each edge has probability $1/|E(H)|$ of being in F'_1, and these probabilities are mutually independent. Similarly, each such edge has probability $1/|E(H)|$ of being in F'_2 and these probabilities are also mutually independent. It follows that in this case, sampling a digraph G from

D_1, and looking at the induced digraph on a set S with the above property, has *exactly* the same distribution as sampling a digraph G from D_2, and looking at the induced digraph on S.

In order to apply Yao's principle and thus complete the proof, we have to show that no deterministic algorithm can distinguish between the distributions D_1 and D_2 with constant probability. To this end, it is clearly enough to show that with probability $1 - o(1)$, any deterministic algorithm that looks at a digraph spanned by less than $(1/\epsilon)^{c' \log 1/\epsilon}$ vertices, has *exactly* the same probability of seeing any digraph regardless of the distribution from which the digraph was chosen. By the discussion in the previous paragraph, this can be proved by establishing that, with high probability, a small set of vertices does not contain three vertices from the same essential copy of H. For a fixed ordered set of three vertices in S, consider the event that they all belong to the same essential copy of H. The first two vertices determine all the vertices of one of these copies uniquely. Now, the conditional probability that the third vertex is also a vertex of the same copy is $h/|V(F)| \leq 1/m$. By the union bound, the probability that the required property is violated is at most

$$|S|^3/m \leq |S|^3 \epsilon^{c \log 1/\epsilon}.$$

This quantity is $o(1)$ as long as $|S| = o((1/\epsilon)^{\frac{c}{3} \log 1/\epsilon})$, where here we applied the lower bound on the size of m given in (3). Therefore, if the algorithm has query complexity $o((1/\epsilon)^{c' \log 1/\epsilon})$ for some absolute positive constant c', it has probability $1 - o(1)$ of looking at a subset on which the distributions D_1 and D_2 are identical, thus, the probability that it distinguishes between D_1 and D_2 is $o(1)$. ■

A slightly more complicated argument than the above can give two distributions D_1 and D_2, such that the graphs in D_1 are *always* ϵ-far from satisfying P^*_H, while the graphs in D_2 always satisfy P^*_H. The idea is to first partition the $m|X|$ essential copies of H into groups of size $|E(H)|$ assuming for simplicity that $|E(H)|$ divides $m|X|$. To create D_1, we randomly pick from each group of $|E(H)|$ copies of H a single copy, and delete all its edges. To create D_2, we do exactly the same as we did in the proof of Theorem 4. It is easy to appropriately modify the proof above in order to show that any deterministic algorithm with query complexity $o((1/\epsilon)^{c \log 1/\epsilon})$ can not distinguish between D_1 and D_2. As this argument has no qualitative advantage, we described the simpler one given above.

6 Concluding Remarks and Open Problems

- As in the case of P_H, there is a huge gap between the general upper bounds for testing P^*_H that were established in [5], and the lower bounds in this paper. It would be very interesting, and probably challenging, to improve any of these bounds. Even in the seemingly simplest case of H being a triangle, we do not know how to improve these bounds.

- Another interesting open problem is to complete the characterizations of easily testable properties P^*_H for undirected graphs H, by solving the cases of $H = P_4, C_4$ (recall that testing the complement of C_4 is equivalent to testing C_4). The case of testing $P^*_{P_4}$ seems the simplest one to resolve, since there are known structural results, that characterize induced P_4-free graphs. These graphs are also known as Complement Reducible graphs, or Cographs for short, and they are precisely the graphs formed from a single vertex under the closure of the operations of union and complement, see [13] and [25]. Cographs arise naturally in such application areas as examination scheduling and automatic clustering of index terms. Cographs have a unique tree representation called a Cotree. It might be possible to use this characterization, and the unique tree representation in order to design an efficient tester for $P^*_{P_4}$.

- Combining Theorem 3 and subsection 4.1, the only unclassified digraphs on 3 vertices are the graph obtained from P_3 by replacing one edge with two anti-parallel edges, and the other by a single edge, and the graph obtained from P_3 by replacing both edges with two anti-parallel edges. As all the digraphs on at least 5 vertices are hard to test, the only remaining unclassified digraphs are the digraphs H on 4 vertices, such that neither H nor $\overline{H}$ contains a triangle, and neither H nor $\overline{H}$ contains a cycle of length 4 that is the core of itself (e.g. the graph obtained from either C_4 or P_4 by replacing each edge with two anti-parallel edges). It will be interesting to classify these cases as well.

- There is an interesting possible connection between the problem of graph isomorphism and testing P^*_H. It is known (see [12]) that for any graph $H \in \{P_2, P_3, P_4, C_4\}$, the graph isomorphism problem can be solved in polynomial time for induced H-free graphs. Moreover, for any other H, any instance of the graph isomorphism problem can be reduced to an instance that is induced H-free. Thus, in some sense, the problem on induced H-free graphs, for H other than P_2, P_3, P_4 and C_4, is *isomorphism hard*. It might be interesting to understand if this connection is indeed meaningful.

References

[1] N. Alon, Testing subgraphs in large graphs, Proc. 42^{nd} IEEE FOCS, IEEE (2001), 434-441.

[2] N. Alon, R. A. Duke, H. Lefmann, V. Rödl and R. Yuster, The algorithmic aspects of the Regularity Lemma, Proc. 33^{rd} IEEE FOCS, Pittsburgh, IEEE (1992), 473-481. Also: J. of Algorithms 16 (1994), 80-109.

[3] N. Alon, W. F. de la Vega, R. Kannan and M. Karpinski, Random Sampling and Approximation of MAX-CSP Problems, Proc. of the 34^{th} ACM STOC, ACM Press (2002), 232-239.

[4] N. Alon, S. Dar, M. Parnas and D. Ron, Testing of clustering, Proc. 41^{st} IEEE FOCS, IEEE (2000), 240-250.

[5] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy, Efficient testing of large graphs, Proc. of 40^{th} FOCS, New York, NY, IEEE (1999), 656–666. Also: Combinatorica 20 (2000), 451-476.

[6] N. Alon, M. Krivelevich, I. Newman and M. Szegedy, Regular languages are testable with a constant number of queries, Proc. 40^{th} FOCS, New York, NY, IEEE (1999), 645–655. Also: SIAM J. on Computing 30 (2001), 1842-1862.

[7] N. Alon and M. Krivelevich, Testing k-colorability, SIAM J. Discrete Math., 15 (2002), 211-227.

[8] N. Alon and A. Shapira, Testing satisfiability, Proc. of the 13^{th} Annual ACM-SIAM SODA, ACM Press (2002), 645-654. Also: Journal of Algorithms, 47 (2003), 87-103.

[9] N. Alon and A. Shapira, Testing subgraphs in directed graphs, Proc. of the 35^{th} Annual Symp. on Theory of Computing (STOC), San Diego, California, 2003, 700–709.

[10] F. A. Behrend, On sets of integers which contain no three terms in arithmetic progression, *Proc. National Academy of Sciences USA* 32 (1946), 331–332.

[11] A. Bogdanov, K. Obata and L. Trevisan, A Lower Bound for Testing 3-Colorability in Bounded-degree Graphs, Proc. 43^{rd} IEEE FOCS, IEEE (2002), 93-102.

[12] D. G. Corneil, H. Lerchs, L. Stewart Burlingham, Complement Reducible Graphs, Discrete Applied Mathematics 3 (1981), 163–174.

[13] D. G. Corneil, Y. Perl and L. K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985), 926–934.

[14] A. Czumaj and C. Sohler, Testing hypergraph coloring, Proc. of ICALP 2001, 493-505.

[15] A. Czumaj and C. Sohler, Property testing in computational geometry, Proceedings of the 8th Annual European Symposium on Algorithms (2000), 155–166.

[16] P. Erdös, P. Frankl and V. Rödl, The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent, Graphs Combin. 2 (1986) 113-121.

[17] E. Fischer, The art of uninformed decisions: A primer to property testing, The Computational Complexity Column of The Bulletin of the European Association for Theoretical Computer Science 75 (2001), 97-126.

[18] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky, Testing juntas, Proc. of The 43^{rd} FOCS (2002), 103-112.

[19] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld and A. Samorodnitsky, Monotonicity testing over general poset domains, Proc. of The 34^{th} STOC (2002), 474-483.

[20] A. Frieze and R. Kannan, Quick approximation to matrices and applications, *Combinatorica* 19 (1999), 175-220.

[21] O. Goldreich, Combinatorial property testing - a survey, In: Randomization Methods in Algorithm Design (P. Pardalos, S. Rajasekaran and J. Rolim eds.), AMS-DIMACS (1998), 45-60.

[22] O. Goldreich, S. Goldwasser and D. Ron, Property testing and its connection to learning and approximation, *Proceedings of the* 37^{th} *Annual IEEE FOCS* (1996), 339–348. Also: JACM, 45 (1998), 653-750.

[23] O. Goldreich and L. Trevisan, Three theorems regarding testing graph properties, Proc. 42^{nd} IEEE FOCS, IEEE (2001), 460-469.

[24] R. L. Graham, B. L. Rothschild and J. H. Spencer, *Ramsey Theory*, Second Edition, Wiley, New York, 1990.

[25] T. A. McKee and F.R. McMorris, **Topics in Intersection Graph Theory**, SIAM, Philadelphia, PA, 1999.

[26] D. Ron, Property testing, in: P. M. Pardalos, S. Rajasekaran, J. Reif and J. D. P. Rolim, editors, *Handbook of Randomized Computing*, Vol. II, Kluwer Academic Publishers, 2001, 597–649.

[27] R. Rubinfeld and M. Sudan, Robust characterization of polynomials with applications to program testing, *SIAM J. on Computing* 25 (1996), 252–271.

[28] E. Szemerédi, Regular partitions of graphs, In: *Proc. Colloque Inter. CNRS* (J. C. Bermond, J. C. Fournier, M. Las Vergnas and D. Sotteau, eds.), 1978, 399–401.

[29] A. C. Yao, Probabilistic computation, towards a unified measure of complexity, Proc. of the 18^{th} IEEE FOCS (1977), 222-227.