Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 104 (2018)

Flexible models for testing graph properties

Oded Goldreich*

June 10, 2018

Abstract

The standard models of testing graph properties postulate that the vertex-set consists of $\{1, 2, ..., n\}$, where n is a natural number that is given explicitly to the tester. Here we suggest more flexible models by postulating that the tester is given access to samples the arbitrary vertex-set; that is, the vertex-set is arbitrary, and the tester is given access to a device that provides uniformly and independently distributed vertices. In addition, the tester may be (explicitly) given partial information regarding the vertex-set (e.g., an approximation of its size).

The flexible models are more adequate for actual applications, and also facilitates the presentation of some theoretical results (e.g., reductions among property testing problems).

This programmatic note contains no real results. It merely presents the suggested definitions and discusses them.

Contents

*Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. Email: oded.goldreich@weizmann.ac.il. This research was partially supported by the Israel Science Foundation (grant No. 671/13).

ISSN 1433-8092

Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a recent textbook [4]). Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by making adequate queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the size of the object).

A significant portion of the foregoing research was devoted to testing graph properties in three different models: the dense graph model (introduced in [6] and reviewed in [4, Chap. 8]), the bounded-degree graph model (introduced in [7] and reviewed in [4, Chap. 9]), and the general graph model (introduced in [12, 11] and reviewed in [4, Chap. 10]). In all these models, it is postulated that the vertex-set consists of $\{1, 2, ..., n\}$, where n is a natural number that is given explicitly to the tester, and this simplified assumption is made in all studies of these models. The simplifying assumption may be imposed, without loss of generality, *provided that (1) the tester can sample the vertex-set, and (2) the tester is explicitly given the size of the vertex-set.*[1]

Having explicitly stated the two foregoing conditions that allow to extend testers of the simplified model to more general settings, we observe that they are of fundamentally different nature. The first condition (i.e., sampleability of the vertex-set) seems essential to testing any non-trivial property, whereas the second condition (i.e., knowledge of the (exact) size of the vertex-set) may be relaxed and even avoided altogether in many cases. For example, the various graph-partition properties and the subgraph-free properties are all testable in a general version of the dense graph model in which only the first condition holds. This is the case since the original testers (presented in [6] and [1], resp.) use the description of the vertex-set only in order to sample it. Needless to say, it follows that the query complexities of these testers are oblivious of the size of the graph (and depend only on the proximity parameter), but (as noted by [2]) the converse does not hold (i.e., testers of size-oblivious query complexity may depend on the size of the graph for their verdict (see also [10])).

Indeed, when the query complexity depends on the size of the graph, the tester need to get at least a sufficiently good approximation of the said size. Typically, such an approximation suffices, as in the case of the bipartite tester for the bounded-degree and general graph models [8, 11]. Hence, we highlight three cases regarding the (a priori) knowledge of the size of the vertex-set (where in all cases the tester is given access to samples drawn from the vertex-set):

1. The tester is explicitly given the exact size of the vertex-set. This ("exact size") case is essentially reducible to the simplified case in which the vertex-set equals $\{1, 2, ..., n\}$ and n is explicitly given to the tester.

2. The tester is explicitly given an approximation of the size of the vertex-set, where the quality of the approximation may vary.

3. The tester is not given explicitly any information regarding the size of the vertex-set.

The foregoing three cases are special cases of a general formulation that supports the study of testing graph properties, where of tested graph has an arbitrary vertex-set, which (w.l.o.g.) is a set of strings.

[1]The latter assertion holds because, as articulated in [9], graph properties are actually properties of unlabeled graphs, and hence testers of such properties may effectively ignore the labels as long as they can sample the vertex-set. Formally, one may consider a bijection from the vertex-set V to $\{1, 2, ..., |V|\}$, and observe that a (general) tester that is given samples of V (and oracle access to a graph with vertex-set V) can construct such a bijection on-the-fly (and emulate a tester of the simplified form). The overhead of this construction is due to the fact that the simplified tester may sample from the set of vertices that it did not encounter so far, whereas the general tester obtains samples that are uniformly distributed in the vertex-set (independent of prior samples and other events in the execution). Hence, the general tester should be able to emulate "samples without repetitions" when given samples with repetitions, but this difference is quite insignificant when the tester has sublinear query complexity.

When testing a graph with vertex-set $V \subset \{0,1\}^*$, the tester is given access to a device that samples uniformly in V. In addition, the tester is explicitly given some information about V, where this information resides in a set of possibilities $p(V)$. The latter formulation allows the given information to be in a predetermined set of possibilities rather than be uniquely determined. For example, the "exact size case" corresponds to $p(V) = \{|V|\}$, the "approximate size case" corresponds to $p(V) = \{n \in \mathbb{N} : n \approx |V|\}$, and the "no information case" corresponds to $p(V) = \{\lambda\}$.

The benefits of the flexible models are two-fold. First, they narrow the gap between the study of testing graph properties and possible real-life applications. Second, they facilitate the presentation of reductions among property testing problems and models, as will be discussed in the sequel. Examples of such reductions include those reviewed in [4, Thm. 9.22] and [4, Thm. 10.4] and those used in [5].

While flexible models may be applicable also to testing properties of objects that are not naturally viewed as graphs, we focus on testing graph properties in the three aforementioned models (i.e., the dense graph model, the bounded-degree graph model, and the general graph model). In all cases we consider only graph properties, which are sets of unlabeled graphs (equiv., set of label graphs that are closed under the renaming of the vertices).[2]

1 Testing Graph Properties in the Dense Graph Model

Here we present a more flexible version of the notion of property testing in the dense graph model (a.k.a the adjacency matrix model, which was introduced in [6] and reviewed in [4, Chap. 8]).

In this model, a graph of the form $G = (V, E)$ is represented by its adjacency predicate $g : V \times V \to \{0,1\}$; that is, $g(u, v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u, v\} \in E$). Distance between graphs (over the same vertex-set) is measured in terms of their foregoing representation; that is, as the fraction of (the number of) entries on which they disagree (over $|V|^2$). The tester is given oracle access to the representation of the input graph (i.e., to the adjacency predicate g) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. In addition, the tester gets some partial information about the vertex-set (i.e., V) as auxiliary input, where this partial information is an element of a set of possibilities denoted $p(V)$. (Indeed, two extreme possibilities are $p(V) = \{V\}$, which is closely related to the standard formulation, and $p(V) = \{\lambda\}$, but we can also consider natural cases such as $p(V) = \{|V|, |V|+1, ..., 2|V|\}$). As usual, the tester is also given the proximity parameter ϵ.

For simplicity (and without loss of generality), we assume that the vertex-set is a set of strings (i.e., a finite subset of $\{0,1\}^*$). Hence, p is a function from sets of strings (representing possible vertex-sets) to sets of strings (representing possible partial information about the vertex-set).

Definition 1.1 (property testing in the dense graph model, revised): *Let Π be a property of graphs and $p : 2^{\{0,1\}^*} \to 2^{\{0,1\}^*}$. A* tester for the graph property Π (in the dense graph model) with partial information *p is a probabilistic oracle machine T that is given access to two oracles, an adjacency predicate $g : V \times V \to \{0,1\}$ and a device denoted $\mathtt{Samp}(V)$ that samples uniformly in V, and satisfies the following two conditions:*

1. *The tester accepts each $G = (V, E) \in \Pi$ with probability at least $2/3$; that is, for every $g : V \times V \to \{0,1\}$ representing a graph in Π and every $i \in p(V)$* (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\mathtt{Samp}(V)}(i, \epsilon) = 1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times V \to \{0,1\}$ that represents a graph that is ϵ-far from Π and $i \in p(V)$, it holds that $\Pr[T^{g,\mathtt{Samp}(V)}(i, \epsilon) = 0] \geq 2/3$, where the graph represented by*

[2]That is, if a graph $G = (V, E)$ has the property, then, for any bijection $\pi : V \to V'$, the graph $G' = (V', \{\{\pi(u), \pi(v)\} : \{u, v\} \in E\}$ has the property.

$g : V \times V \to \{0,1\}$ *is* ϵ-far from Π *if for every* $g' : V \times V \to \{0,1\}$ *that represents a graph in* Π *it holds that* $|\{(u,v) \in V^2 : g(u,v) \neq g'(u,v)\}| > \epsilon \cdot |V|^2$.

The tester is said to have one-sided error probability *if it always accepts graphs in* Π*; that is, for every* $g : V \times V \to \{0,1\}$ *representing a graph in* Π (and every $i \in p(V)$ and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\mathtt{Samp}(V)}(i,\epsilon)=1] = 1$.

The case of $p(V) = \{V\}$ corresponds to the standard model in which one typically postulates that $V = \{1, 2, ..., |V|\}$. This is the case because, given V, the tester may use a bijection between V and $\{1, 2, ..., |V|\}$. The case of $p(V) = \{|V|\}$ is closely related to these cases, except that in this case the bijection can only be constructed on-the-fly. In order to formally state this correspondence, we need to define the query complexity of a tester as in Definition 1.1. For our purposes, it suffice to define the query complexity of the tester as the total number of queries it makes to both its oracles (i.e., the adjacency predicate and the sampling oracles).[3]

Observation 1.2 (the "exact size case" reduces to the standard case): *Suppose that the graph property* Π *has a tester of query complexity* $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ *in the dense graph model under its standard formulation* (e.g., as in [4, Def. 8.2]). *Then,* Π *has a tester of query complexity* $q' = \widetilde{O}(q)$ *in the dense graph model with partial information* p *such that* $p(V) = \{|V|\}$. *Furthermore, one-sided error is preserved, and* $q'(n,\epsilon) = O(q(n,\epsilon))$ *whenever either* $q(n,\epsilon) < n/2$ *or* $q(n,\epsilon) > 4n \ln n$. *The same holds* (simultaneously) *for time complexity.*

Proof Sketch: As outlined in Footnote 1, when testing graph properties, the actual labels of the vertices are immaterial. What matters is whether or not vertices that appear in the current query have appeared in previous queries. Hence, when emulating a tester of the standard formulation, we need to assign new vertices (that appear in queries of this tester) to new vertices in the actual input graph. Specifically, our emulation of the tester T proceeds as follows, where π denotes a partial bijection of $[|V|]$ to V.

- On input $(|V|, \epsilon)$, we invoke T on this very input, while initializing π to be totally undefined.

  (Recall that T issues queries to a adjacency predicate defined over $[|V|] \times [|V|]$.)

- When T issues a query $(u,v) \in [|V|]^2$, we check if $\pi(u)$ and $\pi(v)$ are defined.
  - If both $\pi(u)$ and $\pi(v)$ are defined, then we make the query $(\pi(u), \pi(v))$ to our input graph $G = (V,E)$, and answer T accordingly.
  - If for $w \in \{u,v\}$, the value $\pi(w)$ is undefined, then we get a new sample $s \in V$ from the sampling device (i.e., $s \leftarrow \mathtt{Samp}(V)$). If $\pi^{-1}(s)$ is undefined, then we define $\pi(w) = s$. Otherwise, we try again, and continue till reaching a total number of $q'/3$ (i.e., $q'/3$ invocations of $\mathtt{Samp}(V)$), where q' is the claimed query complexity. Once $\pi(u)$ and $\pi(v)$ are both defined, we proceed as in the previous case.

- If we reached the claimed query complexity (i.e., q') and T has not terminated, then we suspend the execution of T and accept. Otherwise, we output the verdict of V.

Note that if $q(|V|,\epsilon) < |V|/2$, then the probability of obtaining a sample $s \leftarrow \mathtt{Samp}(V)$ on which π^{-1} is undefined is at least $1/2$. On the other hand, $(4n \ln n)/3$ samples of $\mathtt{Samp}(V)$ are very likely to cover all of V. Hence, in all cases, we suspend the execution with very small probability. The foregoing tester can be efficiently implemented (wrt time complexity) by maintaining dynamic sets of the values on which π and π^{-1} are defined. ■

[3] A more refined definition, following [3], may consider the number of queries to each of the oracles. In such a case, it makes sense to refer to the number of queries to the adjacency predicate (resp., the sampling device) as the query (resp., sample) complexity of the tester.

The no-information case. We mention that the testers for the various graph-partition problems presented in [6] satisfy the requirements of Definition 1.1 with $p(V) = \{\lambda\}$ (i.e., the "no partial information" case). Indeed, these (low complexity) testers use the description of the vertex-set only in order to sample it, and so this auxiliary input can be replaced (in them) by a vertex sampling device. The same holds for many other testers (in the dense graph model), including the subgraph-freeness testers presented in [1]. On the other hand, applying the transformation of [9, Thm. 2] to a tester that satisfies Definition 1.1 with $p(V) = \{\lambda\}$, yields a canonical tester of the same type; that is, the auxiliary property that the induced subgraph should satisfy is oblivious of the size of the input graph (cf., [9, 10]). We mention that this special case of Definition 1.1 (i.e., with $p(V) = \{\lambda\}$) is pivotal to the reduction used in the proof of [5, Thm. 4.5].[4]

2 Testing Graph Properties in the Bounded-Degree Graph Model

Here we present a more flexible version of the notion of property testing in the dense graph model (a.k.a the bounded incidence lists model, which was introduced in [7] and reviewed in [4, Chap. 9]).

The bounded-degree graph model refers to a fixed (constant) degree bound, denoted $d \geq 2$. In this model, a graph $G = (V, E)$ of maximum degree d is represented by the function $g : V \times [d] \to V \cup \{\perp\}$ such that $g(v, j) = u \in V$ if u is the j^{th} neighbor of v and $g(v, j) = \perp \notin V$ if v has less than j neighbors.[5] Distance between graphs is measured in terms of their foregoing representation; that is, as the fraction of (the number of) different array entries (over $d|V|$).

As in the dense graph model, the tester is given oracle access to the representation of the input graph (i.e., to the incidence function g) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. In addition, the tester gets some partial information about the vertex-set (i.e., V) as auxiliary input, where this partial information is an element of a set of possibilities denoted $p(V)$. (Again, two extreme possibilities are $p(V) = \{V\}$, which is closely related to the standard formulation, and $p(V) = \{\lambda\}$, but we can also consider natural cases such as $p(V) = \{|V|, |V| + 1, ..., 2|V|\}$). As usual, the tester is also given the proximity parameter ϵ. Again, we assume that the vertex-set is a set of strings (i.e., a finite subset of $\{0, 1\}^*$).

Definition 2.1 (property testing in the bounded-degree graph model, revised): *For a fixed $d \in \mathbb{N}$, let Π be a property of graphs of degree at most d, and $p : 2^{\{0,1\}^*} \to 2^{\{0,1\}^*}$. A* tester for the graph property Π (in the bounded-degree graph model) with partial information p *is a probabilistic oracle machine T that is given access to two oracles, an incidence function $g : V \times [d] \to V \cup \{\perp\}$ and a device denoted* $\texttt{Samp}(V)$ *that samples uniformly in V, and satisfies the following two conditions:*

1. *The tester accepts each $G = (V, E) \in \Pi$ with probability at least $2/3$; that is, for every $g : V \times [d] \to V \cup \{\perp\}$ representing a graph in Π and every $i \in p(V)$* (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\texttt{Samp}(V)}(i, \epsilon) = 1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times [d] \to V \cup \{\perp\}$ that represents a graph that is ϵ-far from Π and $i \in p(V)$, it holds that $\Pr[T^{g,\texttt{Samp}(V)}(i, \epsilon) = 0] \geq 2/3$, where the graph represented by $g : V \times [d] \to V \cup \{\perp\}$ is* ϵ-far from Π *if for every $g' : V \times [d] \to V \cup \{\perp\}$ that represents a graph in Π it holds that $|\{(v, j) \in V \times [d] : g(v, j) \neq g'(v, j)\}| > \epsilon \cdot d|V|$.*

The tester is said to have one-sided error probability *if it always accepts graphs in Π; that is, for every $g : V \times [d] \to V \cup \{\perp\}$ representing a graph in Π* (and every $i \in p(V)$ and $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\texttt{Samp}(V)}(i, \epsilon) = 1] = 1$.

[4] Indeed, this special case of Definition 1.1 appears as [5, Def. 4.3], and triggered us to write the current note.

[5] For simplicity, we adopt the standard convention by which the neighbors of v appear in arbitrary order in the sequence $g(v, 1), ..., g(v, \deg(v))$, where $\deg(v) \stackrel{\text{def}}{=} |\{j \in [d] : g(v, j) \neq \perp\}|$.

Defining the query complexity as in the previous section, we make analogous observations regarding the cases of $p(V) = \{V\}$ and $p(V) = \{|V|\}$. In particular,

Observation 2.2 (the "exact size case" reduces to the standard case): *Suppose that the graph property* Π *has a tester of query complexity* $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ *in the bounded-degree graph model under its standard formulation* (e.g., as in [4, Def. 9.1]). *Then,* Π *has a tester of query complexity* $q' = \widetilde{O}(q)$ *in the bounded-degree graph model with partial information* p *such that* $p(V) = \{|V|\}$. *Furthermore, one-sided error is preserved, and* $q'(n,\epsilon) = O(q(n,\epsilon))$ *whenever* $q(n,\epsilon) < n/3$. *The same holds* (simultaneously) *for time complexity.*

Proof Sketch: We follow the proof of Observation 1.2, except that here "new vertices" are such that have not appeared in previous queries or previous answers (to such queries). Furthermore, when we answer a query $(v,j) \in [|V|] \times [d]$ of the standard tester T by making the query $(\pi(v), j)$ to our own input graph, we may obtain as an answer either an old or a new vertex, denoted $\alpha \in \{0,1\}^*$. In the former case, the value of $\pi^{-1}(\alpha) \in [|V|]$ is already defined, and we provide this value as answer. Otherwise, we answer with a random $w \in [|V|]$ such that $\pi(w)$ is yet undefined, and set $\pi(w) = \alpha$. ■

The cases of no-information and approximate-size. As in the dense graph model, natural testers that have query complexity that depends only on the proximity parameter are easily adapted to the bounded-degree graph model with no partial information (i.e., p such that $p(V) = \{\lambda\}$). The list includes testers that operate by local searchers (reviewed in [4, Sec. 9.2]) and testers that operate by constructing and utilizing partition oracles (reviewed in [4, Sec. 9.5]). Obviously, testers of query complexity that depend on the size of the graph must obtain some information regarding this size, and a constant-factor approximation will typically do (see, e.g., the bipartite tester of [8]). We mention that testers of query complexity that is at least the square root of the size of the graph can obtain such an approximation by sampling the vertex-set, but this method does not preserve one-sided error probability (and only yield probabilistic bounds on the complexity).[6]

The approximate-size version of Definition 2.1 is implicit in the reduction that underlies the presentation of the proof of [4, Thm. 9.22]. The original presentation lacks this notion of a reduction, and so it proceeds by emulating a specific tester for bipartiteness (i.e., the one of [8]) on an auxiliary graph that is derived from the input graph. Using Definition 2.1, we can now say that (one-sided error) testing of cycle-freeness in the bounded-degree graph model is randomly reducible to (one-sided error) testing of bipartiteness in the model of Definition 2.1 with $p(V) = \{\Omega(|V|), ..., O(|V|)\}$. The same holds also w.r.t the proof of [4, Thm. 10.4], which can be presented as a reduction of testing bipartitness in the general graph model to testing bipartiteness in the model of Definition 2.1 with $p(V) = \{\Omega(|V|), ..., O(|V|)\}$.

3 Testing Graph Properties in the General Graph Model

Here we present a more flexible version of the notion of property testing in the general graph model (which was introduced in [12, 11] and reviewed in [4, Chap. 10]). Unlike in the previous two models, here the representation of the graph is decoupled from the definition of the (*relative*) distance between graphs. Following the discussion in [4, Sec. 10.1.2], we define the relative distance between $G = (V,E)$ and $G' = (V,E')$ as the ratio of the symmetric difference of E and E' over $\max(|E|, |E'|) + |V|$.

In this model, a graph $G = (V,E)$ is redundantly represented by both its incidence function $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ (alternatively, we may consider $g_1 : V \times [2|V|] \to V \cup \{\perp\}$)[7] and its adjacency

[6]The obvious procedure is to keep sampling till seeing, say, 100 pairwise collisions, and then outputting the square of the number of trials (divided by 200).

[7]In a previous version of this note, we considered $g_1 : V \times [|V|] \to V \cup \{\perp\}$, where $|V|-1$ served as a trivial degree bound. In retrospect, we feel that using such an upper bound is problematic, since it may allow the tester to determine the number

predicate $g_2 : V \times V \to \{0,1\}$; indeed, as before, $g_1(v,j) = u \in V$ if u is the $j^{\rm th}$ neighbor of v (and $g_1(v,j) = \perp$ if v has less than j neighbors), and $g_2(u,v) = 1$ if and only if $\{u,v\} \in E$. The tester is given oracle access to the two representations of the input graph (i.e., to the functions g_1 and g_2) as well as to a device that returns uniformly distributed elements in the graph's vertex-set. In addition, the tester gets some partial information about the vertex-set (i.e., V) as auxiliary input, where this partial information is an element of a set of possibilities denoted $p(V)$. (Again, two extreme possibilities are $p(V) = \{V\}$, which is closely related to the standard formulation, and $p(V) = \{\lambda\}$, but we can also consider natural cases such as $p(V) = \{|V|, |V|+1, ..., 2|V|\}$). As usual, the tester is also given the proximity parameter ϵ. Again, we assume that the vertex-set is a set of strings (i.e., a finite subset of $\{0,1\}^*$).

Definition 3.1 (property testing in the general graph model, revised):[8] *Let Π be a property of graphs and $p : 2^{\{0,1\}^*} \to 2^{\{0,1\}^*}$. A* tester for the graph property Π (in the general graph model) with partial information *p is a probabilistic oracle machine T that is given access to three oracles, an incidence function $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$, an adjacency predicate $g_2 : V \times V \to \{0,1\}$, and a device denoted $\mathtt{Samp}(V)$ that samples uniformly in V, and satisfies the following two conditions:*

1. *The tester accepts each $G = (V,E) \in \Pi$ with probability at least $2/3$; that is, for every $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ and $g_2 : V \times V \to V \cup \{\perp\}$ representing a graph in Π and every $i \in p(V)$* (and $\epsilon > 0$)*, it holds that $\Pr[T^{g_1,g_2,\mathtt{Samp}(V)}(i,\epsilon){=}1] \geq 2/3$.*

2. *Given $\epsilon > 0$ and oracle access to any G that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and (g_1, g_2) such that $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ $g_2 : V \times V \to V \cup \{\perp\}$ represent a graph that is ϵ-far from Π, and every $i \in p(V)$, it holds that $\Pr[T^{g_1,g_2,\mathtt{Samp}(V)}(i,\epsilon){=}0] \geq 2/3$, where the graph $G = (V,E)$ is* ϵ-far from Π *if for every $G' = (V,E')$ that represents a graph in Π it holds that the symmetric difference of E and E' is greater than $\epsilon \cdot (\max(|E|,|E'|) + |V|)$.*

The tester is said to have one-sided error probability *if it always accepts graphs in Π.*

Defining the query complexity as in the previous sections, we make analogous observations regarding the cases of $p(V) = \{V\}$ and $p(V) = \{|V|\}$. In particular,

Observation 3.2 (the "exact size case" reduces to the standard case): *Suppose that the graph property Π has a tester of query complexity $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ in the general graph model under its standard formulation* (e.g., as in [4, Def. 10.2])*. Then, Π has a tester of query complexity $q' = \widetilde{O}(q)$ in the general graph model with partial information p such that $p(V) = \{|V|\}$. Furthermore, one-sided error is preserved, and $q'(n,\epsilon) = O(q(n,\epsilon))$ whenever $q(n,\epsilon) < n/3$. The same holds* (simultaneously) *for time complexity.*

References

[1] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy. Efficient Testing of Large Graphs. *Combinatorica*, Vol. 20, pages 451–476, 2000.

[2] N. Alon and A. Shapira. A Separation Theorem in Property Testing. *Combinatorica*, Vol. 28 (3), pages 261–281, 2008.

of vertices in the graph (assuming that querying g_1 on an input that is not in its domain results in a suitable indication). On the other hand, allowing an infinite representation of finite graphs is not problematic, since the representation is not used as a basis for the definition of the relative distance between graphs.

[8] Here we follow [4, Def. 10.2], rather than [4, Def. 10.1]. See discussion in [4, Sec. 10.1.2].

[3] M. Balcan, E. Blais, A. Blum, and L. Yang. Active property testing. In *53rd FOCS*, pages 21–30, 2012.

[4] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[5] O. Goldreich. Hierarchy Theorems for Testing Properties in Size-Oblivious Query Complexity. *ECCC*, TR18-098, 2018.

[6] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998.

[7] O. Goldreich and D. Ron. Property Testing in Bounded Degree Graphs. *Algorithmica*, Vol. 32 (2), pages 302–343, 2002.

[8] O. Goldreich and D. Ron. A Sublinear Bipartitness Tester for Bounded Degree Graphs. *Combinatorica*, Vol. 19 (3), pages 335–373, 1999.

[9] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003.

[10] O. Goldreich and L. Trevisan. Errata to [9]. Manuscript, August 2005. Available from `http://www.wisdom.weizmann.ac.il/~oded/p_ttt.html`

[11] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. *SIAM Journal on Computing*, Vol. 33 (6), pages 1441–1483, 2004.

[12] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, Vol. 20 (2), pages 165–183, 2002.

[13] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il