Flexible Models for Testing Graph Properties

Oded Goldreich

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).

An early version of this work appeared as TR18-104 of *ECCC*. The presentation was elaborated in the current revision.

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 has been devoted to testing graph properties in three different models: the dense graph model (introduced in [7] and reviewed in [4, Chap. 8]), the bounded-degree graph model (introduced in [8] and reviewed in [4, Chap. 9]), and the general graph model (introduced in [12,13] 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 employed, 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]

[1] See Observations 1.2, 2.2 and 3.2.

O. Goldreich (Ed.): Computational Complexity and Property Testing, LNCS 12050, pp. 352–362, 2020.
https://doi.org/10.1007/978-3-030-43662-9_19

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, all graph-partition properties (see [7]) and subgraph-free properties are 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 [7] 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 observed 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 [11])).

On the other hand, when the query complexity depends on the size of the graph, the tester needs to obtain 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 [9,12]. 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.
   As shown in Observations 1.2, 2.2 and 3.2, 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 may be employed in the study of testing graph properties in settings in which the tested graph has an arbitrary vertex-set, which (w.l.o.g.) is a set of strings.

The Flexible Models at a Glance. The general formulation postulates that, 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, denoted $p(V)$. The point is that the "given information about V" is allowed 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 $p(V) \in \{n \in \mathbb{N} : n \approx |V|\}$, and the "no information case" corresponds to $p(V) = \{\lambda\}$. This general formulation is called flexible, since it allows its user to determine the function $p : 2^{\{0,1\}^*} \to 2^{\{0,1\}^*}$ according to the setting at hand.

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]

Subsequent Work. In subsequent work [6], the foregoing flexible models were used as a starting point for more general models in which the tester obtains samples that are arbitrarily distributed in the vertex-set. In this case, the definition of the distance between graphs is modified to reflect this vertex distribution; such a modification is not required in the current paper.

Organization. The following three sections present flexible versions of the three aforementioned models of testing graph properties. In Sect. 1 we consider the dense graph model (a.k.a. the adjacency matrix model), in Sect. 2 we consider the bounded-degree graph model (a.k.a. the bounded incidence lists model), and in Sect. 3 we consider the general graph model. The definitional parts of these sections contain some repetitions in order to enable reading them independently of one another.

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 [7] and reviewed in [4, Chap. 8]). The standard version (e.g., as in [4, Def. 8.2]) is obtained as a special case by setting $V = \{1, 2, ..., n\}$ and $p(V) = n$.

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. As usual, the tester is also given the proximity parameter ϵ, which determined when graphs are considered "far apart" (i.e., see the notion of ϵ-far).

[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.

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 $p(V) \in \{|V|, |V|+1, ..., 2|V|\}$.

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 graph $G = (V,E) \in \Pi$ 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 graph 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 $g : V \times V \to \{0,1\}$ 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*

[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.

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* (as in Definition 1.1) *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 $q(n,\epsilon) < n/3$ or $q(n,\epsilon) > 4n \ln n$. The same holds* (simultaneously) *for time complexity.*

Proof: As articulated in [10], 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. Hence, 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.

Note that a tester under the standard formulation may easily generate *new* vertices, since the vertex-set equals $[n] \stackrel{\text{def}}{=} \{1, 2, ..., n\}$ and it is explicitly given n. In contrast, the tester that we construct is given $|V|$, but has no other *a priori* information regarding V. Its only way of obtaining *new* vertices is to query $\texttt{Samp}(V)$ for a sample. The overhead of the transformation is due to the fact that obtaining a *new* vertex may require several samples, when the number of queries made so far (by the original tester) is large (i.e., $\Omega(\sqrt{|V|})$). Yet, this is a minor problem that is easily resolved.

Formally, we emulate a tester T of the standard formulation, by invoking this tester, on input (n,ϵ), where $n = |V|$. We answer T's queries by constructing on-the-fly a bijection π between $[n]$ and the (unknown to us) vertex-set V, and querying our own oracle on the corresponding vertex pairs. Specifically, the query $(u,v) \in [n] \times [n]$ is answered by making a corresponding query $(\pi(u), \pi(v))$ such that if π is not defined on $w \in \{u,v\}$, then we assign $\pi(w)$ a new vertex obtained by sampling V (i.e., we repeatedly invoke $\texttt{Samp}(V)$ till we obtain a vertex that is not in the (defined so far) image of π). Hence, 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 the standard tester T on this very input, while initializing π to be totally undefined.
  (Recall that T issues queries to an adjacency predicate that is defined over $[|V|] \times [|V|]$.)
- When T issues a query $(u,v) \in [|V|]^2$, we check if $\pi(u)$ and $\pi(v)$ are already defined.
  - If both $\pi(u)$ and $\pi(v)$ are already 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., $s \leftarrow \texttt{Samp}(V)$). If $\pi^{-1}(s)$ is undefined, then we define $\pi(w) = s$. Otherwise, we try again, and continue trying till reaching a total number of $q'/3$ (i.e., $q'/3$ invocations of $\texttt{Samp}(V)$), where $q' = \widetilde{O}(q(|V|,\epsilon))$ and q is the query complexity of T. 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 T.

Note that if $q(|V|,\epsilon) < |V|/3$, then the probability of obtaining a sample $s \leftarrow \texttt{Samp}(V)$ on which π^{-1} is undefined is at least $1/3$, since the number of vertices in V that were assigned in response to prior queries of T is less than $2 \cdot q(|V|,\epsilon)$. Hence, in this case, with very high probability, we can obtain $2 \cdot q(|V|,\epsilon)$ distinct elements of V by invoking $\texttt{Samp}(V)$ for $O(q(|V|,\epsilon)$ times, which means that we do not suspend the execution of T. On the other hand, $(4|V|\ln|V|)/3$ samples of $\texttt{Samp}(V)$ are very likely to cover all of V, so we are fine also in case $q(|V|,\epsilon) \geq |V|/3$. Hence, in both cases, we suspend the execution with very small probability. Our choice to accept in the rare case of suspended executions preserves the one-sided error of T (but slightly increases the error on graphs that are far from Π).

Note that the foregoing tester can be efficiently implemented (w.r.t time complexity) by maintaining dynamic sets (of the values) on which π and π^{-1} are defined. ■

The No-Information Case. We mention that the testers for the various graph-partition problems presented in [7] 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].

We also observe that applying the transformation of [10, 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., [10,11]). That is:

Observation 1.3 (canonical testers for the "no partial information case"): *Suppose that Π has a tester of query complexity $q : (0,1] \to \mathbb{N}$ in the dense graph model* (of Definition 1.1) *with no partial information* (i.e., $p(V) = \{\lambda\}$). *Then, there exists a graph property Π' and a tester that, on input ϵ, accepts if and only if the subgraph induces by $O(q(\epsilon))$ random vertices is in Π'.*

We mention that this special case of Definition 1.1 (i.e., $p(V) = \{\lambda\}$) is pivotal to the reduction used in the proof of [5, Thm. 4.5]. In fact, this special case of Definition 1.1 appears as [5, Def. 4.3], and triggered us to write the current paper.

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 bounded-degree graph model (a.k.a. the bounded incidence lists model, which was introduced in [8] and reviewed in [4, Chap. 9]). The standard version (e.g., as in [4, Def. 9.1]) is obtained as a special case by setting $V = \{1,2,...,n\}$ and $p(V) = n$.

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 incidence 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.[4] 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. As usual, the tester is also given the proximity parameter ϵ. 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 $p(V) \in \{|V|, |V| + 1, ..., 2|V|\}$). 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* $\mathtt{Samp}(V)$ *that samples uniformly in V, and satisfies the following two conditions:*

1. *The tester accepts each graph $G = (V, E) \in \Pi$ 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* $\Pr[T^{g,\mathtt{Samp}(V)}(i, \epsilon) = 1] \geq 2/3$.
2. *Given $\epsilon > 0$ and oracle access to any graph 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* $\Pr[T^{g,\mathtt{Samp}(V)}(i, \epsilon) = 0] \geq 2/3$, *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* $\Pr[T^{g,\mathtt{Samp}(V)}(i, \epsilon) = 1] = 1$.

Defining the query complexity as in Sect. 1, we make analogous observations regarding the cases of $p(V) = \{V\}$ and $p(V) = \{|V|\}$. In particular,

[4] 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 $\deg(v) \stackrel{\text{def}}{=} |\{j \in [d] : g(v, j) \neq \perp\}|$.

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* (as in Definition 2.1) *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 in 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 $\alpha \in V$. 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 $\pi(w) = \alpha$. Indeed, this new vertex w is obtained from the sampling device, and this may require repeated sampling as in the proof of Observation 1.2. ■

The No-Information Case. 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 (as in Definition 2.1) 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])[5] and testers that operate by constructing and utilizing partition oracles (reviewed in [4, Sec. 9.5]).

The Approximate-Size Case. 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 [9]). 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 [9]) on an auxiliary graph that is derived from the input graph. Using Definition 2.1, we can now say that

[5] In some cases (e.g. [4, Sec. 9.2.3]), these testers use an estimate of $|V|$ in order to avoided pathological problems that arise when $|V| < B \stackrel{\text{def}}{=} O(1/\epsilon)$. But determining whether not $|V| < B$ holds can be done by using $\widetilde{O}(1/\epsilon)$ samples of V, let alone that it actually suffices to distinguish between $|V| < B$ and $|V| \geq 2B$.

[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).

(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 [12,13] and reviewed in [4, Chap. 10]). The standard version (e.g., as in [4, Def. 10.2]) is obtained as a special case by setting $V = \{1, 2, ..., n\}$ and $p(V) = n$.

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 $g_1 : V \times [b(|V|)] \to V \cup \{\perp\}$, where $b : \mathbb{N} \to \mathbb{N}$ is some degree-bounding function)[7] and its adjacency predicate $g_2 : V \times V \to \{0, 1\}$; indeed, as before, $g_1(v, j) = u \in V$ if u is the j^{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. As usual, the tester is also given the proximity parameter ϵ.

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 $p(V) \in \{|V|, |V| + 1, ..., 2|V|\}$). 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

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

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 graph $G = (V,E) \in \Pi$ 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 graph 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 $g_1 : V \times \mathbb{N} \to V \cup \{\perp\}$ and $g_2 : V \times V \to V \cup \{\perp\}$ 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* (as in Definition 3.1) *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. Alon, N., Fischer, E., Krivelevich, M., Szegedy, M.: Efficient testing of large graphs. Combinatorica **20**, 451–476 (2000)
2. Alon, N., Shapira, A.: A separation theorem in property testing. Combinatorica **28**(3), 261–281 (2008)
3. Balcan, M., Blais, E., Blum, A., Yang, L.: Active property testing. In: 53rd FOCS, pp. 21–30 (2012)
4. Goldreich, O.: Introduction to Property Testing. Cambridge University Press, Cambridge (2017)
5. Goldreich, O.: Hierarchy theorems for testing properties in size-oblivious query complexity. Comput. Complex. **28**(4), 709–747 (2019). https://doi.org/10.1007/s00037-019-00187-2
6. Goldreich, O.: Testing graphs in vertex-distribution-free models. In: 51st STOC, pp. 527–534 (2019)
7. Goldreich, O., Goldwasser, S., Ron, D.: Property testing and its connection to learning and approximation. J. ACM **45**, 653–750 (1998)

8. Goldreich, O., Ron, D.: Property testing in bounded degree graphs. Algorithmica **32**(2), 302–343 (2002)
9. Goldreich, O., Ron, D.: A sublinear bipartitness tester for bounded degree graphs. Combinatorica **19**(3), 335–373 (1999)
10. Goldreich, O., Trevisan, L.: Three theorems regarding testing graph properties. Random Struct. Algorithms **23**(1), 23–57 (2003)
11. Goldreich, O., Trevisan, L.: Errata to [10]. Manuscript, August 2005. http://www.wisdom.weizmann.ac.il/~oded/p_ttt.html
12. Kaufman, T., Krivelevich, M., Ron, D.: Tight bounds for testing bipartiteness in general graphs. SIAM J. Comput. **33**(6), 1441–1483 (2004)
13. Parnas, M., Ron, D.: Testing the diameter of graphs. Random Struct. Algorithms **20**(2), 165–183 (2002)
14. Rubinfeld, R., Sudan, M.: Robust characterization of polynomials with applications to program testing. SIAM J. Comput. **25**(2), 252–271 (1996)