Proximity Oblivious Testing and the Role of Invariances

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL.
oded.goldreich@weizmann.ac.il

Tali Kaufman†
Department of Computer Science
Weizmann Institute of Science
Rehovot, ISRAEL.
kaufmant@mit.edu

April 12, 2010

Abstract

We present a general notion of properties that are characterized by local conditions that are invariant under a sufficiently rich class of symmetries. Our framework generalizes two popular models of testing graph properties as well as the algebraic invariances studied by Kaufman and Sudan (STOC'08).

We show that, in the aforementioned models of testing graph properties, characterization by such invariant local conditions is closely related to proximity oblivious testing (as defined by Goldreich and Ron, STOC'09). In contrast to this relation, we show that, in general, characterization by invariant local conditions is neither necessary nor sufficient for proximity oblivious testing. Furthermore, we show that easy testability is *not* guaranteed even when the property is characterized by local conditions that are invariant under a 1-transitive group of permutations.

Keywords: Property testing, Graph Properties, Locally Testable Codes, Sparse Linear Codes, The Long-Code,

*Partially supported by the Israel Science Foundation (grant No. 1041/08).
†Partially supported by a Koshland Fellowship.

Contents

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a couple of recent surveys [14, 15]). 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 length of the object).

While a host of fascinating results and techniques has emerged, the desire for a comprehensive understanding of what makes some properties easy to test (while others are hard to test) is far from being satisfied.[1] Two general approaches that seem to have a potential of addressing the question (of "what makes testing possible") were suggested recently.

1. Restricting attention to the class of *proximity oblivious testers*, which are constant-query testers that reject any object with probability proportional (but not necessarily linearly proportional) to its distance from the predetermined property. Indeed, the characterization of proximity oblivious testers, in two central models of graph properties, obtained in [8], seems to answer the foregoing question: *graph properties have proximity oblivious testers if and only if they can be characterized in terms of adequate local conditions.*[2]

2. But even before [8], an approach based on *adequately invariant local conditions* was put forward in [12]. It was shown that in the context of testing algebraic properties, *a sufficient condition for testability* (which in fact yields proximity oblivious testers) *is that the property can be characterized in terms of local conditions that are invariant in an adequate sense.*

Thus, these two approaches have a very similar flavor, but they are very different at the actual details. On the one hand, the definition of *proximity oblivious testers* does not refer to any structure of the underlying domain of functions, and the local conditions in the two graph models do not refer explicitly to any invariance. However, invariance under relabeling of the graph's vertices is implicit in the entire study of graph properties (since the latter are defined in terms of such invariance). On the other hand, the *linear invariances* considered in [12] presume that the functions' domain can be associated with some vector space and that the properties are invariant under linear transformations of this vector space.

Thus, the first task that we undertake is providing a definition of a general notion of "characterization by invariant local conditions", where at the very minimum this general definition should unify the notions underlying [8, 12]. Such a definition is presented in Section 2.

Given such a definition, a natural conjecture that arises, hereafter referred to as the invariance conjecture, is that a property has a constant-query proximity-oblivious tester if and only if it can be characterized by invariant local conditions. This conjecture is rigorously formulated within our definitional framework (see Section 2.2) and the current work is devoted to its study. The main results of our study may be stated informally as follows:

1. The invariance conjecture holds in the context of testing graph properties in the dense graph model (see Theorem 3.1).

[1] This assertion is not meant to undermine significant successes of several characterization projects, most notably the result of [1].

[2] We warn the the picture is actually not that clean, because in the case of the bounded-degree model the notion of adequacy includes some technical condition, termed non-propagation.

2. The invariance conjecture holds in the context of testing graph properties in the bounded-degree graph model if and only if all local properties are non-propagating (see Theorem 3.1 and Open Problem 5.8 in [8]).

3. In general, the invariance conjecture fails in both directions.

   (a) Characterization by invariant local conditions is not necessary for proximity oblivious testing. This is demonstrated both by linear properties (see Theorem 5.1) and by the dictatorship property (see Theorem 5.2).

   (b) Characterization by invariant local conditions is not sufficient for proximity oblivious testing (see Theorem 5.3). This is demonstrated by the property called Eulerian orientation (which refers to the orientation of the edges of a cyclic grid, cf. [5]).

Thus, there are natural settings in which the invariance conjecture holds, but there are also natural settings in which it fails (in each of the possible directions).

The technical angle. Items 1 and 2 are established by relying on corresponding results of [8], while our contribution is in observing that the local conditions stated in [8] (in terms of subgraph freeness) coincide with local conditions that are invariant under graph isomorphisms. Actually, to rule out characterizations by other possible invariances, we use the canonization technique of [9, Thm. 2]. In the two examples of Item 3a we rely on the fact that these properties were shown to have (proximity oblivious) testers in [11] and [3], respectively. Thus, in both cases, our contribution is showing that these properties cannot be characterized by invariant local conditions. In Item 3b we rely on a lower bound established in [5] (regarding testing Eulerian orientations of cyclic grids), and our contribution is in observing that this property can be characterized by invariant local conditions.

We mention that the property used towards establishing Item 3b is invariant under a 1-transitive[3] permutation group. Thus, even such an invariance feature does not guarantee easy testability (i.e., a standard tester of query complexity that only depends on the proximity parameter).

Terminology. Throughout the text, when we say proximity oblivious testing we actually mean proximity oblivious testing *in a constant number of queries*. The definition of proximity oblivious testing appears in the appendix.

Organization. In Section 2 we provide a definitional framework that captures the foregoing discussion. In particular, this framework includes a general definition of the notion of characterizations by invariant local conditions and a formal statement of the invariance conjecture. In Section 3 we show the the invariance conjecture holds in the context of testing graph properties in the dense graph model, and in Section 4 we present an analogous conditional (or partial) result for the bounded-degree graph model. The failure of the invariance conjecture is demonstrated in Section 5, and possible conclusions are discussed in Section 6.

[3] A permutation group G over D is called 1-transitive if for every $e, e' \in D$ there exists a $\pi \in G$ such that $\pi(e) = e'$.

2 General Framework

For simplicity, we consider properties of finite functions defined over a finite domain D and having a finite range R, whereas an asymptotic treatment requires considering properties that are infinite sequences of such properties (i.e., a sequence of the type $(\mathrm{P}_n)_{n\in\mathbb{N}}$ where P_n is a set of functions from D_n to R_n). Still, we shall just write P, D, R, and (in order for our asymptotic statements to make sense) one should think of P_n, D_n, R_n. In particular, when we say that some quantity is a "constant", we actually think of D as growing (along with P and possibly R), while the said quantity remains fixed. Thus, in the rest of our presentation, D and R should be considered as generic sets having a variable size, although they will be often omitted from definitions and notations.

The simplified form of the invariant condition. We start by outlining a simplified version of the condition that we seek, regarding a property P (of functions $D \to R$):

1. P is closed under the action of some permutation group G, which is defined over D, and
2. P has a characterization via a constant number of "generic" constraints of constant size such that a function f is in P iff all actual constraints obtained by having G act on the generic constraints are satisfied.

In other words, P can be characterized by a set of constraints that are generated by some permutation group G acting on a constant number of constant-size constraints.

We stress that the foregoing permutation group G is chosen arbitrarily, and may depend on P (and not only on a natural class of properties to which P belongs). Thus, if P is a graph property, then G need not be the group that preserves *all* graph properties (i.e., the vertex-relabeling group), but rather may be any group that extend the vertex-relabeling group. For example, if P is the property of having more edges than non-edges, then the group may be the symmetric group of all (unordered) vertex pairs, which in particular contains the vertex-relabeling group as a subgroup.

2.1 Characterization by generated constraints

We now generalize and clarify the above discussion. First we need to define what we mean by a constraint. A constraint will be a pair consisting of domain elements and a Boolean predicate applied to the corresponding values, and it is satisfied by a function f if applying the predicate to the f-values at the specified locations yields the Boolean value 1 (representing `true`).

Definition 2.1 (constraints): *A* constraint *is a pair* $((e_1, ..., e_c), \phi)$ *such that* $e_1, ..., e_c$ *are distinct elements in* D*, and* $\phi : R^c \to \{0,1\}$ *is an arbitrary predicate. We say that the foregoing is a constraint of arity* c (or a c-constraint). *A function* $f : D \to R$ *is said to satisfy the foregoing constraint if* $\phi(f(e_1), ..., f(e_c)) = 1$.

Note that at this point the predicate ϕ may depend on the sequence of elements $(e_1, ..., e_c)$. Such a dependence will not exist in the case that a large set of constraints is generated based on few constraints (as in Definition 2.3).

The next notion is of characterization by a set of constraints. A property P of functions is characterized by a set of constraints if f is in P if and only f satisfies all constraints in the set.

Definition 2.2 (characterization by constraints): *Let* C *be a set of constraints and* P *be a property. We say that* P *is* characterized *by* C *if for every* $f : D \to R$ *it holds that* $f \in \mathrm{P}$ *if and only if* f *satisfies each constraint in* C.

Next, we consider the set of constraints generated by the combination of (1) a fixed set of constraints, (2) a group of permutations over D, and (3) a group of permutations over R. For starters, the reader is advised to think of the second group as of the trivial group containing only the identity permutation. In general, we shall consider a subset of the set of all pairs consisting of a permutation as in (2) and a permutation as in (3).

Definition 2.3 (generated constraints): *Let C be a finite set of c-constraints, and M be a set of pairs consisting of a permutation over D and a permutation over R* (i.e., for any $(\pi,\mu) \in M$ it holds that π is a permutation of D and μ is a permutation R). *The set of* constraints generated *by C and M, denoted $\mathtt{CONS}(C,M)$, is defined by*

$$\mathtt{CONS}(C,M) \stackrel{\text{def}}{=} \{((\pi(e_1),...,\pi(e_c)),\phi\circ\mu^{-1}) : ((e_1,...,e_c),\phi)\in C\ ,\ (\pi,\mu)\in M\} \tag{1}$$

where $\phi\circ\mu^{-1}(v_1,...,v_c)$ denotes $\phi(\mu^{-1}(v_1),...,\mu^{-1}(v_c))$.

Note that saying that f satisfies $((\pi(e_1),...,\pi(e_c)),\phi\circ\mu^{-1})$ means that

$$(\phi\circ\mu^{-1})(f(\pi(e_1)),...,f(\pi(e_c))) = \phi(\mu^{-1}(f(\pi(e_1))),...,\mu^{-1}(f(\pi(e_c)))) = 1,$$

which means that $\mu^{-1}\circ f\circ\pi$ satisfies the constraint $((e_1,...,e_c),\phi)$. Regarding the use of $\mu^{-1}\circ f\circ\pi$ rather than $\mu\circ f\circ\pi$, see discussion following Definition 2.5.

Notation: As in Definition 2.3, it will be convenient to generalize functions to sequences over their domain. That is, for any function F defined over some set S, and for any $e_1,...,e_t \in S$, we denote the sequence $(F(e_1),...,F(e_t))$ by $F(e_1,...,e_t)$. Throughout the text, `id` will be used to denote the identity permutation, where the domain is understood from the context.

2.2 The invariance condition

Returning to the condition outlined initially, let us now formulate it as follows. We consider a group of pairs (π,μ) such that π is a permutation over D and μ is a permutation over R with a group operation that corresponds to component-wise composition of permutations (i.e., $(\pi_1,\mu_1)\odot(\pi_2,\mu_2) = (\pi_1\circ\pi_2,\mu_1\circ\mu_2)$, where $\odot$ denotes the group operation). We call such a group a group of permutation pairs, and note that it need not be a direct product of a group of permutation over D and a group of permutations over R.

Definition 2.4 (the invariance condition): *A property* P satisfies the invariance condition *if there exists a constant, denoted c, a finite set of c-constraints, denoted C, and a group, denoted M, of permutation pairs over $D\times R$ such that* P *is characterized by $\mathtt{CONS}(C,M)$. In this case, we also say that* P satisfies the invariance condition w.r.t M.

Recall that the group operation $\odot$ of M satisfies $(\pi_1,\mu_1)\odot(\pi_2,\mu_2) = (\pi_1\circ\pi_2,\mu_1\circ\mu_2)$, where $\circ$ denotes composition of permutations. Thus, M induces a permutation group over D (as well as one over R), but M is not necessarily their direct product (e.g., for $D = R$, it may be that $M = \{(\pi,\pi) : \pi \in G\}$, where G is a permutation group over D).

The invariance condition and covering the domain. *We confine our discussion to the case that the domain contains only elements that are influential w.r.t the property* P; that is, for every $e \in D$, there exists $f_1 \in \mathrm{P}$ and $f_0 \notin \mathrm{P}$ such that $f_1(x) = f_0(x)$ for every $x \in D \setminus \{e\}$. Observe that if property P satisfies the invariance condition w.r.t M, then M induces a transitive permutation group on a constant fraction of D. This follows because the permutation group (over D) induced by M must map a constant number of elements (i.e., those appearing in the constraint set C) to all elements of D.

The main question. We ask what is the relation between satisfying the invariance condition and having a proximity oblivious tester (of constant-query complexity). One natural conjecture, hereafter referred to as the invariance conjecture, is that *a property satisfies the invariance condition if and only if it has a proximity oblivious tester.* Weaker forms of this conjecture refer to its validity within various models of property testing. This leads us to ask what are "models of property testing".

2.3 Models of property testing

Natural model of property testing can be defined by specifying the domain and range of functions (i.e., D and R) as well as closure features of the properties in the model.[4] We elaborate below (and mention that this view was elaborated independent by Sudan [17]).

For example, the model of testing graph properties in the adjacency matrix representation, introduced in [6], refers to $D = \binom{[N]}{2}$ and $R = \{0,1\}$ as well as to the permutation group over D that is defined by all relabeling of $[N]$. Specifically, an N-vertex graph is represented by the Boolean function $g : \binom{[N]}{2} \to \{0,1\}$ such that $g(\{u,v\}) = 1$ if and only if u and v are adjacent in the graph. Here an adequate closure feature gives rise to graph properties, where P is a graph property if, for every such function g, and every permutation ψ over $[N]$, it holds that $g \in \mathrm{P}$ iff $g_\psi \in \mathrm{P}$, where $g_\psi(\{u,v\}) \stackrel{\mathrm{def}}{=} g(\{\psi(u),\psi(v)\})$.

In general, closure features are defined by groups of pairs of permutations, just as those in Definition 2.4.

Definition 2.5 (closure features): *Let M be as in Definition 2.4. We say that a property* P is closed under M *if, for every $(\pi,\mu) \in M$, it holds that $f \in \mathrm{P}$ if and only if $\mu \circ f \circ \pi^{-1} \in \mathrm{P}$.*

Note that $\mu \circ f \circ \pi^{-1}$ (rather than $\mu \circ f \circ \pi$) is indeed the natural choice, since f maps D to R whereas the new function $f' = \mu \circ f \circ \pi^{-1}$ is meant to map $\pi(D)$ to $\mu(R)$; thus, when f' is applied to $e' = \pi(e)$ this results in first recovering e, next applying f, and finally applying μ.

Definition 2.6 (closure-based models of property testing): *The* model of M *consists of the class of all properties that are closed under M.*

For example, the model of testing graph properties in the adjacency matrix representation corresponds to the set M that equals all pairs $(\pi, \mathtt{id})$ such that there exists a permutation ψ over $[N]$ such that $\pi(\{u,v\}) = \{\psi(u),\psi(v)\}$ (for all $\{u,v\} \in D = \binom{[N]}{2}$). As we shall see, not all "common models of property testing" can be reduced to Definition 2.6, but nevertheless Definition 2.6 is a good starting point; that is, various models can be naturally defined as subclasses of the class of

[4] In addition, one may consider sub-models that are obtained by requiring the functions in such a model to satisfy some auxiliary properties.

all properties that are closed under some group M (where typically in such cases the subclass are characterized by a set of constraints that are generated as in Definition 2.3).[5]

We observe that closure under M is a necessary condition for satisfying the invariance condition with respect to M.

Proposition 2.7 *If* P *satisfies the invariance condition w.r.t* M*, then* P *is closed under* M.

Proof: For any $f \in \mathrm{P}$ and $(\pi_0, \mu_0) \in M$, consider $f' \stackrel{\text{def}}{=} \mu_0 \circ f \circ \pi_0^{-1}$. We shall show that $f \in \mathrm{P}$ if and only if $f' \in \mathrm{P}$. Suppose that P is characterized by $\mathtt{CONS}(C, M)$, and consider an arbitrary constraint in $\mathtt{CONS}(C, M)$. By definition (of being generated from (C, M)), this constraint has the form $(\pi(e_1), ..., \pi(e_c)), \phi \circ \mu^{-1})$, where $((e_1, ..., e_c), \phi) \in C$ and $(\pi, \mu) \in M$. Our aim is to show that f' satisfies this constraint if and only if f satisfies some related constraint in $\mathtt{CONS}(C, M)$, where the two constraints are related via (π_0, μ_0).

We start by looking at the value of $(\phi \circ \mu^{-1})(f'(\pi(e_1)), ..., f'(\pi(e_c)))$, which we shorthand as $(\phi \circ \mu^{-1})(f'(\pi(e_1, ..., e_c)))$. Plugging-in the definition of f', what we now look at is $(\phi \circ \mu^{-1})((\mu_0 \circ f \circ \pi_0^{-1})(\pi(e_1, ..., e_c)))$, which may be written as $\phi(\mu^{-1} \circ \mu_0 \circ f \circ \pi_0^{-1} \circ \pi(e_1, ..., e_c))$, which in turn equals $\phi((\mu^{-1} \circ \mu_0) \circ f \circ (\pi_0^{-1} \circ \pi)(e_1, ..., e_c))$. That is, we consider whether f satisfies the constraint $((\pi_0^{-1} \circ \pi)(e_1, ..., e_c), \phi \circ (\mu^{-1} \circ \mu_0))$, which can be written as $((\pi_0^{-1} \circ \pi)(e_1, ..., e_c), \phi \circ (\mu_0^{-1} \circ \mu)^{-1})$. But this constraint is in $\mathtt{CONS}(C, M)$, since it is generated from $((e_1, ..., e_c), \phi) \in C$ by using the pair $(\pi_0^{-1} \circ \pi, \mu_0^{-1} \circ \mu) \in M$. Thus, f' satisfies the constraint generated (from $((e_1, ..., e_c), \phi)$) by $(\pi_0^{-1} \circ \pi, \mu_0^{-1} \circ \mu)$ if and only if f satisfies the constraint generated (from it) by (π, μ). It follows that f' satisfies all constraints in $\mathtt{CONS}(C, M)$ if and only if f satisfies all constraints in $\mathtt{CONS}(C, M)$. ■

3 The Invariance Conjecture holds in the Dense Graph Model

We prove the invariance conjecture holds in the special case of graph properties in the adjacency matrix representation model (a.k.a the dense graph model). Recall that in the adjacency matrix model, an N-vertex graph is represented by the (symmetric) Boolean function $g : [N] \times [N] \to \{0, 1\}$ such that $g(u, v) = 1$ if and only if u and v are adjacent in the graph.

We rely on a recent result of [8], which states that (in this model) P has a proximity oblivious tester if and only if it is a subgraph-freeness property. We next observe that being a subgraph-freeness property is equivalent to satisfying invariance condition *with respect to the canonical set*, where a set M is canonical if $M = M' \times \{\mathrm{id}\}$ such that M' is the group of permutations over vertex-pairs that is induced by vertex-relabeling. (Indeed, the canonical set is the very set that defines the current model; see Section 2.3). So it is left to show that P *satisfies the invariance condition if and only if* P *satisfies the invariance condition with respect to the canonical set.* We thus get

Theorem 3.1 *Suppose that* P *is a set of Boolean functions over the set of unordered pairs over* $[N]$ *such that* P *is closed under relabeling of the base set* (i.e., P is a graph property that refers to the adjacency representation of graphs). *Then,* P *has a proximity oblivious tester if and only if* P *satisfies the invariance condition. Furthermore, if* P *satisfies the invariance condition, then it satisfies this condition with the canonical set.*

[5] Indeed, an alternative formulation of the model of testing graph properties in the adjacency matrix representation is obtained by starting from $D = [N] \times [N]$ and M that equals all pairs $(\pi, \mathtt{id})$ such that $\pi(u, v) = (\psi(u), \psi(v))$, for some permutation ψ over $[N]$ (and all $(u, v) \in D = [N] \times [N]$). In such a case, we consider the subclass of symmetric function (i.e., functions g such that $g(u, v) = g(v, u)$ for all $(u, v) \in D$).

Proof: The key observation is that, in this model, *a property satisfies the invariance condition with respect to the canonical set if and only if it is a subgraph-freeness property*, where throughout this proof subgraph-freeness means not having certain induced graphs (which are specified in a forbidden set). The backward direction (i.e., from subgraph-freeness to the invariance condition) follows by observing that every subgraph-freeness property satisfies the invariance condition with respect to the canonical set, because it can be generated by the predicate that forbids certain unlabeled graphs (e.g., not having $F = ([n], E_F)$ as an induced subgraph is captured by the constraint $((\{1,2\},..,\{1,n\},...,\{n-1,n\}),\phi)$ such that $\phi(a_{1,2},...,a_{n-1,n}) = 1$ if and only if F is not represented by $(a_{i,j})_{i,j}$). In proving the other direction (i.e., from the invariance condition to subgraph-freeness), observe that the "base" constraints may be viewed as a predicate on an unlabeled induced subgraph; that is, the constraint $((\{i_1,j_1\},..,\{i_c,j_c\}),\phi)$ can be viewed as forbidding all induced subgraphs that are consistent with some $(a_{i_k,j_k})_{k\in[c]}$ such that $\phi(a_{i_1,j_1},...,a_{i_c,j_c}) = 0$.

Another important observation is that *if* P *satisfies the invariance condition then it does so with the canonical pair*. This observation is proven as follows. Let P be characterized by $\texttt{CONS}(C,M)$, where M is not necessarily the canonical set. Then, we view $\texttt{CONS}(C,M)$ (or rather the uniform distribution over it) as a ((possibly "weak") non-adaptive) tester with one-sided error; that is, this tester always accepts any graph in P and its error probability (on no-instances) is strictly less than 1 (i.e., it accepts graphs that are not in P with probability is at most $1 - |\texttt{CONS}(C,M)|^{-1}$). Applying [9, Thm. 2], we obtain a tester with similar one-sided error that only inspects the graph induced by a random constant-size vertex-set. (Indeed, the transformation in [9, Thm. 2] preserves the detection probability no matter how small it is.) The latter tester gives rise to a characterization of P that can be generated by the decision predicate of this tester coupled with the the group of vertex-relabeling; that is, P satisfies the invariance condition with the canonical set.

The current theorem now follows by combining the two foregoing observations with [8, Thm. 4.7]. Specifically, by [8, Thm. 4.7], P *has a proximity oblivious tester, if and only if it is a subgraph freeness property*, By the first observation, P is a subgraph freeness property if and only if P satisfies the invariance condition with the canonical set, whereas (by the second observation) P satisfies the invariance condition if and only if P satisfies the invariance condition with the canonical set. ■

4 The Invariance Conjecture in the Bounded-Degree Graph Model

The next natural challenge is proving a result analogous to Theorem 3.1 for the bounded-degree graph model (introduced in [7]). Unfortunately, only a partial result is established here, because of a difficulty that arises in [8, Sec. 5] (regarding "non-propagation"), to be discussed below.

But first, we have to address a more basic difficulty that refers to fitting the bounded-degree graph model within our framework (i.e., Section 2.3). Recall that the standard presentation of the bounded-degree model represents a N-vertex graph of maximum degree d by a function $g : [N] \times [d] \to \{0,1,...,N\}$ such that $g(v,i) = u \in [N]$ if u is the i^{th} neighbor of v and $g(v,i) = 0$ if v has less than i neighbors. This creates technical difficulties, which can be resolved in various ways.[6] The solution adopted here is to modify the representation of the bounded-degree graph model such that N-vertex graphs are represented by functions from $[N]$ to subsets of $[N]$. Specifically, such a graph is represented by a function $g : [N] \to 2^{[N]}$ such that $g(v)$ is the set of neighbors of vertex v.

[6]The problem is that here it is important to follow the standard convention of allowing the neighbors of each vertex to appear in arbitrary order (as this will happen under relabeling of vertex names), but this must allow us to permute over $[d]$ without distinguishing vertices from the 0-symbol. One possibility is to give up the standard convention by which the vertices appear first and 0-symbols appear at the end of the list. We choose a different alternative.

Furthermore, we are only interested in functions g that described undirected graphs, which means that $g : [N] \to 2^{[N]}$ should satisfy $u \in g(v)$ iff $v \in g(u)$ (for every $u, v \in [N]$).

Theorem 4.1 *Suppose that* P *is a set of functions from* $[N]$ *to* $\{S \subset [N] : |S| \leq d\}$ *that corresponds to undirected graph properties; in particular,* P *is closed under the following canonical set* M_0 *defined by* $(\pi, \mu) \in M_0$ *if and only if* π *is a permutation over* $[N]$ *and* μ *acts analogously on sets* (i.e., $\mu(S) = \{\pi(v) : v \in S\}$).[7] *Then:*

1. *If* P *has a proximity oblivious tester, then it satisfies the invariance condition.*
2. *If* P *satisfies the invariance condition, then it satisfies it with respect to the canonical set, and it follows that* P *is a generalized subgraph freeness property* (as defined in [8, Def. 5.1]).

Recall that by [8, Sec. 5], if P is a generalized subgraph freeness property *that is non-propagating*, then P has a proximity oblivious tester. But it is unknown whether each generalized subgraph freeness property is non-propagating. (We note that this difficulty holds even with respect to properties that satisfies the invariance condition with respect to the canonical set.)[8]

Proof: As in the dense graph model (i.e., Theorem 3.1), the key observation is that a property in this model satisfies the invariance condition with respect to the canonical set if and only if it is a generalized subgraph-freeness property (as defined in [8, Def. 5.1]). Thus, Part (1) follows immediately from [8, Thm. 5.5], and the point is proving Part (2).[9]

Suppose that P is characterized by $\texttt{CONS}(C, M)$. Viewing the uniform distribution over $\texttt{CONS}(C, M)$ as a (very weak) one-sided error non-adaptive tester, we apply a "canonicalization" procedure that is analogous to [9, Thm. 2], and obtain a (very weak) tester that inspects the neighborhoods of c randomly distributed vertices. This yields a characterization of P by $\texttt{CONS}(\{((1, ..., c), \phi)\}, M_0)$, where ϕ is this tester's decision predicate. So we are done. ■

5 The Invariance Conjecture Fails in Some Cases

We show that, in general, the invariance condition is neither necessary nor sufficient for the existence of proximity oblivious testers (POTs).

5.1 The Invariance Condition is not necessary for POT

We present two examples (i.e., properties) that demonstrate that satisfying the invariance condition is not necessary for having a proximity oblivious tester. Both examples are based on sparse linear codes that have (proximity oblivious) codeword tests (i.e., these codes are locally testable). In both cases, the key observation is that satisfying the invariance condition with respect to M (as in Definition 2.4) requires that M is "rich enough" since the domain permutations should map a

[7] Recall that we also assume that for every $g \in$ P it holds that $u \in g(v)$ iff $v \in g(u)$ (for every $u, v \in [N]$). We note that this extra property is easy to test.

[8] In fact, the negative example in [8, Prop. 5.4] can arise in our context. Specifically, consider the set of constraints generated by the constraint $((1, 2), \phi)$ such that $\phi(S_1, S_2) = 1$ iff both (1) $|\{i \in \{1, 2\} : S_i = \emptyset\}| \neq 1$ and (2) $|S_1| \in \{0\} \cup \{2i - 1 : i \in \mathsf{N}\}$. (Indeed, condition (1) mandates that if the graph contains an isolated vertex then it contains no edges, whereas condition (2) mandates that all non-isolated vertices have odd degree.)

[9] The point (i.e., Part (2)) is showing that if P satisfies the invariance condition, then it satisfies it with respect to the canonical set. We mention that the transformation from the possibly adaptive character of a proximity oblivious tester to the non-adaptive character of the invariance condition (equivalently, generalized subgraph-freeness) is performed in [8, Thm. 5.5].

fixed number of elements to all the domain elements. On the other hand, Proposition 2.7 requires that the property be closed under M, whereas this is shown to be impossible in both examples. In the first example, presented next, the property will be shown to be closed only under the trivial pair $(\mathtt{id}, \mathtt{id})$.

Theorem 5.1 *There exists a property, denoted* P*, of Boolean functions such that* P *has a proximity oblivious tester but does not satisfies the invariance condition. Furthermore,* P *is a linear property; that is, if* $f_1, f_2 \in \mathrm{P}$ *then* $f_1 + f_2 \in \mathrm{P}$*, where* $(f_1 + f_2)(x) = f_1(x) \oplus f_2(x)$ *for every* x*.*

Proof: We consider a random linear property of dimension $\ell = O(\log n)$. That is, for uniformly selected functions $g_1, ..., g_\ell : [n] \to \{0,1\}$, we consider the property $\mathrm{P}_n = \{\sum_{i \in I} g_i : I \subseteq [\ell]\}$. Actually, we repeat this selection for every value of n, obtaining the property $\mathrm{P} = (\mathrm{P}_n)_{n \in \mathbb{N}}$. It was shown in [11] that, with high probability over these random choices, the property P has a POT. We shall show that, with high probability over these random choices, the property P does not satisfy the invariance condition.

The key observation is that satisfying the invariance condition with respect to M (as in Definition 2.4) requires that M is non-trivial (i.e., contains a non-trivial pair), because otherwise P_n is characterized by a fixed (i.e., independent of n) number of constraints (which is highly improbable for random g_i's). On the other hand, Proposition 2.7 requires that P_n be closed under M, which is highly improbable when M is non-trivial. Specifically, we will show that with high probability (over the choice of P_n), for every non-trivial (π, μ), there exists $f \in \mathrm{P}_n$ such that $\mu \circ f \circ \pi^{-1} \notin \mathrm{P}$. We distinguish two cases: (1) the case that π is not the identity permutation but μ is the identity permutation, and (2) the case that μ is not the identity permutation (which implies that $\mu(b) = 1-b$ for every $b \in \{0,1\}$).

Claim 5.1.1 *Let* π *be a permutation such that* $m \stackrel{\mathrm{def}}{=} |\{i \in [n] : \pi(i) \neq i\}| > 0$*. Then, for a random* P_n*, the probability that* $\{f \circ \pi : f \in \mathrm{P}_n\} = \mathrm{P}_n$ *is less than* $2^{-m\ell/4}$*.*

Note that the number of permutations that satisfy the hypothesis is smaller than $\binom{n}{m} \cdot (m!) < 2^{m \log_2 n}$. Thus, the aggregated probability for the aforementioned Case (1) is a small constant (i.e., $\sum_{m>0} 2^{-m \cdot ((\ell/4) - \log_2 n)}$ is smaller than, say, 0.01).

Proof: As a warm-up we upper bound the probability that $g \circ \pi = g$, where $g : [n] \to \{0,1\}$ is uniformly distributed. For $g \circ \pi = g$ to hold, g must be constant on each cycle of π. Denoting the number of cycles by $c \leq m/2$, it follows that $\Pr_g[g \circ \pi = g] = 2^{-m+c} \leq 2^{-m/2}$. The argument extends to the case that we wish $g \circ \pi = g + f$ to hold for an arbitrary fixed f and a random g. Specifically, consider a cycle of π, denoted $i_1, ..., i_t$. Then, $\Pr_g[(\forall j \in [t-1])\ g(j+1) = g(j) + f(j)] = 2^{-(t-1)}$. It is even easier to prove that $\Pr_g[g \circ \pi = f] \leq 2^{-m/2}$, since actually $\Pr_g[g \circ \pi = f] = 2^{-n}$. We now turn to upper-bound the probability that $\{f \circ \pi : f \in \mathrm{P}_n\} = \mathrm{P}_n$, by upper-bounding

$$
\begin{aligned}
\Pr_{g_1,...,g_\ell}[(\forall i \in [\ell])\ g_i \circ \pi \in \mathrm{P}_n] &= \Pr_{g_1,...,g_\ell}\left[\forall i \in [\ell]\ \exists I_i \subseteq [\ell] \text{ s.t. } g_i \circ \pi = \sum_{j \in I_i} g_j\right] && (2) \\
&\leq \sum_{I_1,...,I_\ell \subseteq [\ell]} \Pr_{g_1,...,g_\ell}\left[\forall i \in [\ell]\ g_i \circ \pi = \sum_{j \in I_i} g_j\right] && (3)
\end{aligned}
$$

We break the sum in Eq. (3) into two parts, separating the single term that corresponds to $(I_1, ..., I_\ell) = (\{1\}, ..., \{\ell\})$ from all other terms. The contribution of the first term to Eq. (3) is upper-bounded by $(2^{-m/2})^\ell$, because $\Pr_{g_1,...,g_\ell}[\forall i \in [\ell]\ g_i \circ \pi = g_i]$ equals $\prod_{i=1}^{\ell} \Pr_{g_i}[g_i \circ \pi = g_i]$. For each other term corresponding to $(I_1, ..., I_\ell) \neq (\{1\}, ..., \{\ell\})$, we pick an arbitrary i such that $I_i \neq \{i\}$, and note that $\Pr_{g_1,...,g_\ell}[g_i \circ \pi = \sum_{j \in I_i} g_j]$ equals 2^{-n}, since g_i is uniformly distributed even when fixing the value of $\sum_{j \in I_i} g_j$. Furthermore, this assertion holds even if we only select g_i and $f_i = \sum_{j \in I_i} g_j$ at random (where in case $I_i = \emptyset$ we mean setting $f_i \equiv 0$). We now consider an iterative process starting with $i_1 = i$, such that at the first step we select uniformly g_{i_1} and $f_{i_1} = \sum_{j \in I_{i_1}} g_j$. Recall that we have $\Pr_{g_{i_1}, f_{i_1}}[g_{i_1} \circ \pi = f_{i_1}] = 2^{-n}$. For $k = 2, ..., \ell/2$, at the k^{th} step we set i_k such that g_{i_k} is independent of $g_{i_1}, ..., g_{i_{k-1}}$ and $f_{i_1}, ..., f_{i_{k-1}}$ (where $f_i = \sum_{j \in I_i} g_j$), and uniformly select g_{i_k} and f_{i_k} (unless f_{i_k} was already determined in which case it is left unchanged). Note that such a i_k exists as long as $k \leq \ell/2$, but I_{i_k} need not be different than $\{i_k\}$. Then, the probability that $g_{i_k} \circ \pi = \sum_{j \in I_{i_k}} g_j$, conditioned on the values of $g_{i_1}, ..., g_{i_{k-1}}$ and $f_{i_1}, ..., f_{i_{k-1}}$, is at most $2^{-m/2}$, where the probability is taken merely over the choice of g_{i_k} (and possibly f_{i_k}). Thus, the contribution of this generic term to Eq. (3) is upper-bounded by $2^{-n} \cdot (2^{-m/2})^{(\ell/2)-1}$. Using the union bound, we upper-bound the contribution of all these $(2^\ell)^\ell - 1$ terms by

$$2^{\ell^2} \cdot 2^{-(n-(m/2))} \cdot (2^{-m/2})^{\ell/2}, \tag{4}$$

which is upper-bounded by $2^{-(m\ell/4)-1}$ (because $2^{\ell^2} \cdot 2^{-(n-(m/2))} < 1/2$). The claim follows (because $2^{-m\ell/2} < 2^{-(m\ell/4)-1}$). □

Claim 5.1.2 *Let $\mu(b) = 1 - b$. Then, for a random P_n, the probability that there exists a permutation π such that $\{\mu \circ f \circ \pi^{-1} : f \in \mathrm{P}_n\} = \mathrm{P}_n$ is negligible as a function of n* (i.e., is vanishes faster than any polynomial fraction (in n)).

Proof: It suffices to show that, while the all-zero function is in P_n, with very high probability the constant-one function is not in P_n. This is the case because, with overwhelmingly high probability, for every non-empty $I \subseteq [\ell]$ it holds that $|\{j \in [n] : \sum_{i \in I} g_i(j) = 1\}|$ is in $(1 \pm o(1)) \cdot n/2$. □

Combining Claims 5.1.1 and 5.1.2, we conclude that with high constant probability P is not closed under any non-trivial pair. Recalling the initial discussion, the theorem follows. ■

Testing the Long-Code (a.k.a dictatorship tests). We refer to the property $\mathrm{P} = (\mathrm{P}_n)$, where for $n = 2^\ell$, it holds that $f : \{0,1\}^\ell \to \{0,1\}$ is in P_n if and only if there exists $i \in [\ell]$ such that $f(\sigma_1 \cdots \sigma_\ell) = \sigma_i$. Such a function f is a dictatorship (determined by bit i) and can be viewed as the i^{th} codeword in the long-code (i.e., the long-code encoding of i). Note that this property is closed under the pair $(\pi, \mathtt{id})$, where π is a permutation π over $\{0,1\}^\ell$, if and only if there exists a permutation ϕ over $[\ell]$ such that $\pi(\sigma_1 \cdots \sigma_\ell) = \sigma_{\phi(1)} \cdots \sigma_{\phi(\ell)}$. (An analogous consideration applies to pairs $(\pi, \mathtt{flip})$, where $\mathtt{flip}(\sigma) = 1 - \sigma$ for every $\sigma \in \{0,1\}$.) We shall show that these are the only pairs under which the dictatorship property is closed, and it will follow that the dictatorship property violates the invariance condition.

Theorem 5.2 *The dictatorship property violates the invariance condition, although it has a proximity oblivious tester.*

Proof: The fact that the dictatorship property has a proximity oblivious tester is established in [3, 13].[10] We shall show that this property violates the invariance condition because it is not

[10] The longcode test of [3] only refers to the case that ℓ is a power of 2.

closed under pairs (π,μ) unless π either preserves the (Hamming) weight of the strings or preserves this weight under flipping.

Indeed, the notion of (Hamming) weight is pivotal to this proof, where the weight of a string $\alpha \in \{0,1\}^\ell$, denoted $\mathtt{wt}(\alpha)$, is defined as the number of bit positions that contain a one (i.e., $\mathtt{wt}(\sigma_1\cdots\sigma_\ell) \stackrel{\rm def}{=} |\{i\in[\ell]:\sigma_i=1\}|$). We first claim that if P_n is closed under (π,μ) then $\mathtt{wt}(\pi(\alpha))$ equals either $\mathtt{wt}(\alpha)$ or $\ell-\mathtt{wt}(\alpha)$ for every $\alpha\in\{0,1\}^\ell$. (These two cases correspond to whether $\mu=\mathtt{id}$ or $\mu=\mathtt{flip}$ (i.e., $\mu(\sigma)=1-\sigma$).)

Suppose that π maps some ℓ-bit string α to a string β that has a different weight (i.e., $\mathtt{wt}(\beta)\neq\mathtt{wt}(\alpha)$). Then, $|\{f\in\mathrm{P}_n : f(\alpha)=1\}| = \mathtt{wt}(\alpha)$, because for every $f\in\mathrm{P}_n$ there exists a different $i\in[\ell]$ such that $f(\sigma_1\cdots\sigma_\ell)=\sigma_i$. Similarly, $|\{f\circ\pi : f\in\mathrm{P}_n \wedge (f\circ\pi)(\alpha)=1\}| = \mathtt{wt}(\beta)$, since $(f\circ\pi)(\alpha)=f(\beta)$. Using $\mathtt{wt}(\alpha)\neq\mathtt{wt}(\beta)$, we infer that $\mathrm{P}_n\neq\{f\circ\pi : f\in\mathrm{P}_n\}$, since each set contains a different number of functions that evaluated to 1 at the point α. This handles the case of $\mu=\mathtt{id}$, and the case of $\mu=\mathtt{flip}$ is handled similarly (i.e., if π maps some ℓ-bit string α to a string β such that $\mathtt{wt}(\beta)\neq\ell-\mathtt{wt}(\alpha)$, then $\mathrm{P}_n\neq\{\mu\circ f\circ\pi : f\in\mathrm{P}_n\}$).

Having established the above, we note that if P had satisfied the invariance condition then the corresponding M would have mapped a fixed number of elements to all domain elements. But this fixed number of domain elements (i.e., ℓ-bit long strings) have a fixed number of weights, whereas (by Proposition 2.7 and the above) the set M may only contain pairs (π,μ) such that π preserves (or "complements") the weight of strings. This contradicts the requirement that all $\ell+1$ different weights must be covered by the generated constraints, and the theorem follows. ■

5.2 The Invariance Condition is not sufficient for POT

We next demonstrate that the invariance condition does not suffice for obtaining a proximity oblivious tester. Actually, this example also shows that the invariance condition does not suffice for the standard definition of testing (with query complexity that only depends on the proximity parameter).

Theorem 5.3 *There exists a property, denoted* P*, of Boolean functions such that* P *satisfies the invariance condition but has no proximity oblivious tester. Furthermore, the invariant condition holds with respect to a single linear constraint that refer to four domain elements, and a group of domain permutations that is 1-transitive. Moreover,* P *cannot be tested* (in the standard sense) *within query complexity that only depends on the proximity parameter.*

Proof: We use a lower bound of [5] that refers to the query complexity of testing Eulerian orientations of fixed (and highly regular) bounded-degree graphs. Specifically, [5, Thm. 9.14] proves an $\Omega(\log\ell)$ query lower bound on the complexity of testing whether the orientation of an ℓ-by-ℓ cyclic grid is Eulerian. It follows that this property has no POT, while we shall see that it satisfies the invariance condition.

We represent the orientation of the ℓ-by-ℓ cyclic grid by two functions $h,v:\mathsf{Z}_\ell\times\mathsf{Z}_\ell\to\{0,1\}$ such that $h(i,j)$ represents the orientation of the horizontal edge between the vertices (i,j) and $(i,j+1)$, whereas $v(i,j)$ represents the orientation of the vertical edge between the vertices (i,j) and $(i+1,j)$, and the arithmetics is of Z_ℓ (i.e., modulo ℓ). Specifically, $h(i,j)=1$ (resp., $v(i,j)=1$) indicates an orientation from (i,j) to $(i,j+1)$ (resp., $(i+1,j)$). (Needless to say, we can pack both functions in a single function; for example, $f(1,i,j)=h(i,j)$ and $f(0,i,j)=v(i,j)$.)

The key observation is that the Eulerian orientation property can be characterized by 4-constraints that are generated from a single constraint. Specifically, this property is characterized by the set of 4-constraints $\{h(i,j)+v(i,j)=h(i,j-1)+v(i-1,j) : i,j \in \mathsf{Z}_\ell\}$, where the constraint $h(i,j)+v(i,j)=h(i,j-1)+v(i-1,j)$ mandates that exactly two of the four edges of vertex (i,j) are oriented outwards. Finally, note that this set of constraints is generated by the single constraint $h(1,1)+v(1,1)=h(1,0)+v(0,1)$ and the set of mappings $\{(\pi_{r,s}, \mathtt{id})\}$, where $\pi_{r,s}(i,j)=(i+r,j+s)$. The main claim follows.

The only part of the furthermore claim that requires elaboration is the claim that the group of domain permutations is 1-transitive. To show this we explicitly consider the packing of the aforementioned two functions in a single function $f:\{0,1\}\times\mathsf{Z}_\ell\times\mathsf{Z}_\ell\to\{0,1\}$ such that $f(1,i,j)=h(i,j)$ and $f(0,i,j)=v(i,j)$. We redefine the domain permutations $\pi_{r,s}$ such that $\pi_{r,s}(\sigma,i,j)=(\sigma,i+r,j+s)$ and introduce an auxiliary permutation π' such that $\pi'(\sigma,i,j)=(1-\sigma,j,i)$. Observe that a generic constraint (now written as $f(1,i,j)+f(0,i,j)=f(1,i,j-1)+f(0,i-1,j)$) is preserved under the auxiliary permutation π'. The full claim now follows. ■

6 Conclusions

While the invariance conjecture holds in two natural models of testing graph properties, it was shown to fail in other settings. These failures, described in Section 5, are of three different types.

1. As shown in Theorem 5.1, proximity oblivious testers exist also for properties that are only closed under the identity mapping. That is, a strong notion of testability is achievable also in the absence of any invariants.

2. As shown in Theorem 5.2, the existence of proximity oblivious testing for properties that do not satisfy the invariance condition is not confined to unnatural properties and/or to properties that lack any invariance.

3. As shown in Theorem 5.3, the invariance condition does not imply the existence of a standard tester of query complexity that only depends on the proximity parameter. (Note that the non-existence of such testers implies the non-existence of proximity oblivious testers.) Furthermore, this holds even if the invariance condition holds with respect to a group of domain permutations that is 1-transitive.

Our feeling is that the fact that the invariance condition is not necessary for proximity oblivious testing is less surprising than the fact that the former is insufficient for the latter. Giving up on the necessity part, we wonder whether a reasonable strengthening of the invariance condition may suffice for proximity oblivious testing.

A natural direction to consider is imposing additional restrictions on the group of domain permutations. As indicated by Theorem 5.3, requiring this group to be 1-transitive does not suffice, and so one is tempted to require this group to be 2-transitive[11] (as indeed suggested in [10] w.r.t standard testing). Recalling that if P is closed under a 2-transitive group (over the domain) then P is self-correctable (and thus consists of functions that are pairwise far apart), one may also wonder about only requiring 1-transitivity but restricting attention to properties that consists of functions that are pairwise far apart. We mention that the property used in the proof of Theorem 5.3 contains functions that are close to one another.

Actually, restricting attention to properties that are closed under a 1-transitive group of domain permutations, we may return to the question of necessity and ask whether the existence of proximity

[11] A permutation group G over D is called 2-transitive if for every $(e_1,e_2),(e_1',e_2')\in\binom{D}{2}$ there exists a $\pi\in G$ such that $\pi(e_1)=e_1'$ and $\pi(e_2)=e_2'$.

oblivious testers in this case implies the invariance condition. Note that our proof of Theorems 5.1 and 5.2 relies on the fact that the corresponding group is not 1-transitive (e.g., in the first case the group action is trivial and in the second case it has a non-constant number of orbits).

Acknowledgments

We are grateful to Dana Ron for useful discussions.

References

[1] N. Alon, E. Fischer, I. Newman, and A. Shapira. A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity. In *38th STOC*, pages 251–260, 2006.

[2] N. Alon and A. Shapira. A Characterization of Easily Testable Induced Subgraphs. *Combinatorics Probability and Computing*, 15:791–805, 2006.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[4] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS*, Vol. 47, No. 3, pages 549–595, 1993.

[5] E. Fischer, O. Lachish, A. Matsliah, I. Newman, and O. Yahalom. On the query complexity of testing orientations for being Eulerian. Full version available from URL `http://www.cs.technion.ac.il/~oyahalom`. To appear in *ACM Trans. on Algorithms.* Extended abstract in the proceedings of *12th RANDOM*, LNCS 5171, pages 402–415, 2008.

[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. On Proximity Oblivious Testing. *ECCC*, TR08-041, 2008. Also in the proceedings of the *41st STOC*, 2009.

[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] E. Grigorescu, T. Kaufman, and M. Sudan. 2-Transitivity is Insufficient for Local Testability. In *23rd CCC*, pages 259–267, 2008.

[11] T. Kaufman and M. Sudan. Sparse Random Linear Codes are Locally Testable and Decodable. In the proceedings of the *48th FOCS*, pages 590–600, 2007.

[12] T. Kaufman and M. Sudan. Algebraic Property Testing: The Role of Invariances. In *40th STOC*, pages 403–412, 2008.

[13] M. Parnas, D. Ron, and A. Samorodnitsky. Testing basic boolean formulae. *SIAM Journal on Discrete Math*, 16(1):20–46, 2002.

[14] D. Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, Vol. 1 (3), pages 307–402, 2008.

[15] D. Ron. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in TCS*, to appear.

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

[17] M. Sudan. Invariance in Property Testing. *ECCC*, TR10-051, 2010.

Appendix: Property testing and Proximity Oblivious Testers

We first recall the standard definition of property testing.

Definition A.1 *Let* $\mathrm{P} = \bigcup_{n\in\mathbb{N}} \mathrm{P}_n$*, where* P_n *contains functions defined over the domain* D_n*. A* tester for a property P *is a probabilistic oracle machine* T *that satisfies the following two conditions:*

1. *The tester accepts each* $f \in \mathrm{P}$ *with probability at least* $2/3$*; that is, for every* $n \in \mathbb{N}$ *and* $f \in \mathrm{P}_n$ (and every $\epsilon > 0$)*, it holds that* $\Pr[T^f(n,\epsilon)=1] \geq 2/3$.
2. *Given* $\epsilon > 0$ *and oracle access to any* f *that is* ϵ*-far from* P*, the tester rejects with probability at least* $2/3$*; that is, for every* $\epsilon > 0$ *and* $n \in \mathbb{N}$*, if* $f : D_n \to R_n$ *is* ϵ*-far from* P_n*, then* $\Pr[T^f(n,\epsilon)=0] \geq 2/3$*, where* g *is* ϵ-far from P_n *if, for every* $g \in \mathrm{P}_n$*, it holds that* $|\{e \in D_n : f(e) \neq g(e)\}| > \epsilon \cdot n$.

If the tester accepts every function in P *with probability 1, then we say that it has* one-sided error*; that is,* T *has one-sided error if for every* $f \in \mathrm{P}$ *and every* $\epsilon > 0$*, it holds that* $\Pr[T^f(n,\epsilon)=1] = 1$*. A tester is called* non-adaptive *if it determines all its queries based solely on its internal coin tosses* (and the parameters n and ϵ)*; otherwise it is called* adaptive.

The query complexity of a tester is measured in terms of the size parameter, n, and the proximity parameter, ϵ. In this paper we focus on the case that the complexity only depends on ϵ (and is independent of n).

Turning to the definition of proximity-oblivious testers, we stress that they differ from standard testers in that they do not get a proximity parameter as input. Consequently, assuming these testers have sublinear complexity, they can only be expected to reject functions not in P with probability that is related to the distance of these functions from P. This is captured by the following definition.

Definition A.2 *Let* $\mathrm{P} = \bigcup_{n\in\mathbb{N}} \mathrm{P}_n$ *be as in Definition A.1. A* proximity-oblivious tester *for* P *is a probabilistic oracle machine* T *that satisfies the following two conditions:*

1. *The machine* T *accepts each function in* P *with probability 1; that is, for every* $n \in \mathbb{N}$ *and* $f \in \mathrm{P}_n$*, it holds that* $\Pr[T^f(n)=1] = 1$.
2. *For some* (monotone) *function* $\rho : (0,1] \to (0,1]$*, each function* $f \notin \mathrm{P}$ *is rejected by* T *with probability at least* $\rho(\delta_{\mathrm{P}}(f))$*, where* $\delta_{\mathrm{P}}(f) \stackrel{\mathrm{def}}{=} \min_{g\in\mathrm{P}}\{\delta(f,g)\}$ *and* $\delta(f,g) \stackrel{\mathrm{def}}{=} \Pr_{x\in\Omega}[f(x) \neq g(x)]$.

The function ρ *is called the* detection probability of the tester T.

In general, the query complexity of a proximity-oblivious tester may depend on the size parameter, n, but in this paper we focus on the case that this complexity is constant.