On the Average-Case Complexity of Property Testing

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

July 13, 2007

Abstract

Motivated by a recent study of Zimand (*22nd CCC*, 2007), we consider the average-case complexity of property testing (focusing, for clarity, on testing properties of Boolean strings). We make two observations:

1. In the context of average-case analysis with respect to the uniform distribution (on all strings of a fixed length), property testing is trivial. Specifically, either the YES-instances (i.e., instances having the property) or the NO-instances (i.e., instances that are far from having the property) are exponentially rare, and thus the tester may just reject (resp., accept) obliviously of the input.
2. Turning to average-case derandomization with respect to distributions that assigns noticeable probability mass to both YES-instances and NO-instances, we identify a natural class of distributions and testers for which average-case derandomization results can be obtained directly (i.e., without using randomness extractors). Furthermore, the resulting deterministic algorithm may preserve the non-adaptivity of the original tester. (In contrast, Zimand's argument utilizes a strong type of randomness extractors and introduces adaptivity into the testing process.)

We also present a natural example for which the approach of Item 2 is inapplicable, while Zimand's approach may be applicable.

Keywords: Property Testing, Average-Case, Derandomization, Randomness Extractors.

Introduction

Recently, Zimand showed that any randomized sublinear-time algorithm can be derandomized yielding deterministic algorithms of polynomially-related complexity that err on a negligible fraction of the instances [Z]. Specifically, he showed that, for some fixed $\alpha > 0$, randomized algorithms of time-complexity T such that $T(n) < n^{\alpha}$ can be emulated by poly(T)-time deterministic algorithms that err on at most an $\exp(-\Omega(T \log T))$ fraction of the instances. Needless to say, Zimand's work (as well as the current note) refers to a "direct access" model of computation in which each bit of the input can be read at unit cost. Zimand noted the relevance of his work to property testing, but our view is that this aspect of his work should be evaluated with great care. Articulating this view is the main motivation of the current note.

Average-case with respect to the uniform distribution

In discussing the theoretical significance of his work, Zimand says "it shows that the properties that can be checked in sublinear time depend, except for a few inputs, on just a few bits of the input and the locations of these bits can be found very fast."[1] We fear that such a phrasing does not put adequate emphasis on the exception clause (i.e., "except for a few inputs"). Furthermore, in our opinion, the crux of property testing is dealing with non-typical (i.e., exceptional) inputs, whereas dealing with random inputs is typically uninteresting.

We first note that average-case analysis with respect to the uniform distribution is not adequate in the context of testing properties of strings, which in turn cover almost all types of property testing problems (e.g., testing graph properties in the adjacency matrix model). The reason being that property testing problems are special type of promise problems[2] in which one should distinguish instances having the property from instances that are far from any string having the property. However, as shown in Section 1, *for every property of n-bit strings either the first set* (i.e., instances having the property) *or the second set* (i.e., instances far from having the property) *has exponentially vanishing density.* In the first (resp., second) case, *a trivial tester* that rejects (resp., accepts) every input (without reading a single bit) *is correct on all but a exponentially vanishing fraction of all inputs*, where the exceptional cases consists of all the YES-instances (resp., all the "far-away" instances).

Indeed, the average-case complexity of promise problems is meaningful only with respect to *distributions that assign noticeable probability mass to both* YES-*instances and* NO-*instances* (because otherwise a trivial algorithm as above will do). However, the uniform distribution cannot satisfy the latter condition in the case of promise problems that correspond to property testing (of Boolean strings).

A direct average-case derandomization for many natural cases

We thus turn to average-case derandomization with respect to distributions that assigns noticeable probability mass to both YES-instances and NO-instances (i.e., "far-away" instances). While Zimand's results may be applicable to this context too[3], we identify a natural class of distributions

[1] See last paragraph of [Z, Sec. 1.0].

[2] Recall that promise problems [ESY] are represented as pairs of non-intersecting sets $A, B \subseteq \{0,1\}^*$ and solving such problems requires distinguishing inputs in A from inputs in B, while an arbitrary answer is allowed for inputs that are neither in A nor in B. For such a promise problem we say that a string in $A \cup B$ satisfy the promise (while strings outside $A \cup B$ violate the promise).

[3] As shown in Section 1, such distributions must have min-entropy at most $n - \Omega(n)$, while [Z] does not provide results for this range of paramters. Still it is possible that the basic approach of [Z] coupled with an adequate

and testers for which average-case derandomization results can be obtained directly (and provide a more illuminating account of what is actually going on).

Recall that, in continuation to [GW], Zimand [Z] emulates the computation of the original randomized tester by applying a (special type of) randomness extractor to the input and replacing the coin tosses of the original tester with corresponding outputs of the extractor. Consequently, even if the original tester is non-adaptive (as is the case with most natural property testers), the resulting deterministic algorithm is adaptive (because the simulation step depends on the bits read in the randomness-extraction step). In contrast, we show that, in many natural cases, *an average-case derandomization can be obtained by arbitrarily fixing the coins of the original tester.*

To illustrate the point, let us consider the problem of testing whether a given Boolean string has a majority of 1-values (or is far from any such string). In this case, we may obtain a deterministic algorithm by inspecting the value of the *first* few bits in the string, where this algorithm decides correctly on almost all n-bit strings that have a number of 1-values that is bounded away from $n/2$; that is, ruling by the majority of the inspected bits, we decide correctly on almost all elements in the set of n-bit strings having Hamming weight outside the interval $[0.49n, 0.51n]$. Furthermore, any fixed set of sufficiently many bit positions can be used for this purpose. For a general treatment, see Section 2.

We illustrate the general treatment by considering the special case of testing graph properties in the adjacency matrix model (as in, e.g., [GGR]). In this setting (but also in other natural settings), the natural property testers use their randomness solely for determining the bit positions to be examined in the input. Furthermore, at the cost of squaring the query complexity, we may assume that any graph property can be tested by using randomness in such a restricted manner [GT]. In Section 2, we show that *a deterministic tester that inspects the subgraph induced by any fixed set of vertices* (of adequate size) *errs rarely with respect to any distribution on labeled graphs that is invariant under isomorphism.*

Final comments

We note that in many cases, it is easier to construct property testers that work only on typical objects drawn from natural distributions rather than construct standard testers that work on all objects. This fact is not reflected by the results shown in Section 2, where we convert standard (randomized) testers into deterministic "average-case testers".

We emphasize the fact that, when considering worst-case complexity, randomness is essential for testing natural properties (see, e.g., [GS], and note that this is an unconditional result). Indeed, this result stand in contrast to the aforementioned average-case testing results, and provides a formal sense in which "average-case testing" is easier than standard (worst-case) testing. However, we claim that things go beyond this sense: detecting random objects that are far from a property is typically easier than detecting arbitrary objects that are far from this property.

Reservations regarding our own opinions. The direct average-case derandomizations presented in Section 2 refer to distributions that are invariant under natural reshuffling of the presentation of the studied objects (e.g., in the case of labeled graph we considered distribution that are invariant under isomorphism). Although such distributions arise naturally in many cases, distributions that lack this feature are natural in other cases. For example, consider a distribution over real-valued vectors (or matrices) that is obtained by the following two-step process: first a vector (resp., a matrix) is selected according to an arbitrary distribution, and then each of its entries is

randomness extractor may be applicable to such distributions.

pertubed at random and independently of anything else. The resulting distribution may not satisfy any of the invariances considered in Section 2, but it does have high min-entropy. Recalling that various natural properties of vectors (resp., matrices) can be tested in probabilistic sublinear time (cf., e.g., [EKKRV, FK]), we note that Zimand's approach [Z] may be applicable in this case (and if so yield average-case derandomization of natural appeal).

1 Average-case with respect to the uniform distribution

We start by recalling the setting of property testing, when specialized to bit strings (of fixed length). We comment that other finite objects can be naturally represented by such generic strings, and thus corresponding properties can be naturally cast in this framework. The most notable example is property testing of graphs in the adjacency matrix model (as introduced in [GGR]).

For a generic length parameter n, we consider the set of all strings over $\{0,1\}^n$, and an arbitrary property $\mathtt{P}_n \subseteq \{0,1\}^n$. Property testing with respect to a distance parameter $\epsilon > 0$ corresponds to distinguishing inputs in $\mathtt{P}_n$ from inputs in $\Gamma_\epsilon(\mathtt{P}_n)$, where

$$\Gamma_\epsilon(\mathtt{P}_n) \stackrel{\rm def}{=} \{x \in \{0,1\}^n : \forall z \in \mathtt{P}_n \;\; \Delta(x,z) > \epsilon \cdot n\} \tag{1}$$

and $\Delta(x_1 \cdots x_n, z_1 \cdots z_n) = |\{i : x_i \neq z_i\}|$ denotes the number of bits on which $x = x_1 \cdots x_n$ and $z = z_1 \cdots z_n$ disagree.[4] That is, property testing with respect to ϵ corresponds to deciding the promise problem $(\mathtt{P}_n, \Gamma_\epsilon(\mathtt{P}_n))$. However, as we shall see, with respect to the uniform distribution on $\{0,1\}^n$, this promise problem is trivial on the average. That is,

Theorem 1.1 ([AS, Thm. 7.5.3], reformulated): *For every constant $\epsilon > 0$ there exists a constant $c > 0$ such that for every n if $|\mathtt{P}_n| \geq 2^{-cn} \cdot 2^n$ then $|\Gamma_\epsilon(\mathtt{P}_n)| \leq 2^{-cn} \cdot 2^n$. More generally, if $|\mathtt{P}_n| \geq \rho \cdot 2^n$ and $\epsilon \geq \sqrt{\frac{8\ln(1/\rho)}{n}}$, then $|\Gamma_\epsilon(\mathtt{P}_n)| \leq \rho \cdot 2^n$.*

Indeed, Theorem 1.1 can be reformulated by referring to a uniformly distributed $x \in \{0,1\}^n$. This reformulation (of the special case of constant $\epsilon > 0$) asserts that (for some constant $c > 0$) either $\Pr_x[x \in \mathtt{P}_n] < 2^{-cn}$ or $\Pr_x[x \in \Gamma_\epsilon(\mathtt{P}_n)] \leq 2^{-cn}$. In the first case, a tester that always reject is correct on all but at most a 2^{-cn} fraction of the n-bit inputs, whereas in the second case the same holds for a tester that always accepts. Thus, property testing is trivial on the average with respect to any distribution that has min-entropy $m \stackrel{\rm def}{=} n - o(n)$ (i.e., a distribution X_n such that of every x it holds that $\Pr[X_n = x] \leq 2^{-m}$).[5]

Proof: The theorem is merely a reformulation of a well-known result regarding the volume of balls around sets. Specifically, let $\mathcal{B}_d(S)$ denote the set of n-bit long strings that are at distance at most d from some string in S (i.e., $\mathcal{B}_d(S) \stackrel{\rm def}{=} \{x \in \{0,1\}^n : \exists y \in S \text{ s.t. } \Delta(x,y) \leq d\}$). Then, Theorem 7.5.3 in [AS] asserts that *if* $|S| \geq e^{-\lambda^2/2} \cdot 2^n$ *then* $|\mathcal{B}_{2\lambda\sqrt{n}}(S)| \geq (1 - e^{-\lambda^2/2}) \cdot 2^n$. Using $S = \mathtt{P}_n$ and $\lambda = \sqrt{2\ln(1/\rho)}$, where $\rho = |\mathtt{P}_n|/2^n$, we get $|\mathcal{B}_{\sqrt{8n\ln(1/\rho)}}(\mathtt{P}_n)| \geq (1-\rho) \cdot 2^n$. Noting that $\Gamma_\epsilon(\mathtt{P}_n) = \{0,1\}^n \setminus \mathcal{B}_{\epsilon n}(\mathtt{P}_n)$, the general claim follows. The special case follows by noting that $\rho = 2^{-cn}$ implies $\sqrt{(8\ln(1/\rho))/n} = \sqrt{8c/\log e}$ (and so using $c = \epsilon^2/8$ will do). ■

[4] An alternative exposition may refer to Boolean functions of the form $f : [n] \to \{0,1\}$. In this case $\Delta(f,g) = |\{i : f(i) \neq g(i)\}|$.

[5] In fact, we may allow min-entropy $m = n - (cn/2)$, where c is a constant as above. For such a distribution X_n (of min-entropy $n - (cn/2)$), it holds that either $\Pr[X_n \in \mathtt{P}_n] \leq 2^{-cn/2}$ or $\Pr[X_n \in \Gamma_\epsilon(\mathtt{P}_n)] \leq 2^{-cn/2}$.

Generalization. We note that Theorem 1.1 generalizes to properties of sequences over any alphabet Σ. That is, for any property $\mathtt{P}_n \subseteq \Sigma^n$, it holds that *if* $|\mathtt{P}_n| \geq \rho \cdot |\Sigma|^n$ *and* $\epsilon \geq \sqrt{\frac{8\ln(1/\rho)}{n}}$, *then* $|\Gamma_\epsilon(\mathtt{P}_n)| \leq \rho \cdot |\Sigma|^n$, where $\Gamma_\epsilon(\mathtt{P}_n)$ denotes the set of n-long sequences over Σ that are ϵ-far from every sequence in $\mathtt{P}_n$. (See further details in the Appendix.)

2 A direct average-case derandomization for many natural cases

In this section we show that, in many interesting settings of property testing, average-case derandomization results can be obtained more directly than by following the approach suggested by Zimand (see Footnote 3). We start by considering the concrete setting of testing graph properties in the adjacency matrix model (as in [GGR]), and later generalize the treatment to other settings. Indeed, the setting of testing graph properties in the adjacency matrix model provides the most appealing application of the general approach to be described later.

Recall that in this model (for testing graph properties), n-vertex graphs are represented by Boolean strings of length n^2. For technical reasons, we prefer to represent such graphs as Boolean functions defined over the set of the $\binom{n}{2}$ (unordered) vertex-pairs, which is actually more natural (as well as non-redundant). Note that the set of all permutations over $[n]$ induces a transitive group of permutations over these pairs, where the permutation $\pi : [n] \to [n]$ induces a permutation that maps pairs of the form $\{i, j\}$ to $\{\pi(i), \pi(j)\}$. Indeed, any graph property is invariant under this group, which is hereafter referred to as the group of vertex-relabeling; that is, $G = ([n], E)$ has the property if and only if $\pi(G) = ([n], \{\{\pi(i), \pi(j)\} : \{i, j\} \in E\})$ has this property.

Theorem 2.1 *Let $\mathtt{G}_n$ be a graph property, referring to n-vertex graphs, and let X_n be any arbitrary distribution of n-vertex graphs that is invariant under the group of vertex-relabeling* (i.e., for every permutation $\pi : [n] \to [n]$ it holds that X_n and $\pi(X_n)$ are identically distributed). *Suppose that the promise problem $(\mathtt{G}_n, \Gamma_\epsilon(\mathtt{G}_n))$ can be decided correctly* (in the worst case) *by a probabilistic tester of query complexity $q(n, \epsilon)$ and error probability at most $1/3$. Then, for every $k < n/O(q(n, \epsilon)^2)$, there exists a deterministic algorithm of query complexity $O(k \cdot q(n, \epsilon)^2)$ that inspects only vertex pairs that correspond to the vertices $1, ..., O(k \cdot q(n, \epsilon))$ and is correct on a random input X_n with probability at least 2^{-k}.*

As will be clear from the proof, we may use any $O(k \cdot q(n, \epsilon))$ fixed vertices rather than the vertex set $\{1, ..., O(k \cdot q(n, \epsilon))\}$.

Proof: By [GT, Thm. 2], we may convert the original tester into a canonical tester that selects uniformly a set of $n' \stackrel{\text{def}}{=} O(q(n, \epsilon))$ vertices, denoted R, and accepts if and only if the subgraph induced by R has some predetermined (graph) property $\mathtt{G}'_{n'}$. By invoking the resulting (canonical) tester $t \stackrel{\text{def}}{=} O(k)$ times, we reduce its (worst-case) error probability to 2^{-k}. We claim that the resulting tester, denoted A, can be derandomized (for average-case performance) by merely using any *fixed* set of $t \cdot n'$ vertices rather than a *random* set of $t \cdot n'$ vertices as selected by A. We denote the resulting deterministic algorithm by D.

To prove the above claim, we consider an arbitrary input graph G that satisfies the promise (i.e., either $G \in \mathtt{G}_n$ or G is ϵ-far from $\mathtt{G}_n$). By the foregoing discussion we know that the probability that A errs on input G is at most 2^{-k}. Let π denote a uniformly distributed permutation of $[n]$, and consider the graph $\pi(G)$ obtained from G by relabeling its vertices according to π. Note that $\pi(G) \in \mathtt{G}_n$ if and only if $G \in \mathtt{G}_n$ (and, likewise, $\pi(G)$ is ϵ-far from $\mathtt{G}_n$ iff G is ϵ-far from $\mathtt{G}_n$). On the other hand, the distribution of the view of A on input G is identical to distribution of the view of D

on input $\pi(G)$, because a random π maps any fixed set of vertices to a uniformly distributed set of vertices. We stress that the first probability space is defined over the coin tosses of A, whereas the second probability space is defined over the random relabeling π. We conclude that the probability that D errs on input $\pi(G)$ is at most 2^{-k}.

By the hypothesis that X_n is invariant under the group of vertex-relabeling, it follows that X_n can be described by a process in which one first selects a random graph G (possibly $G \leftarrow X_n$) and then outputs $\pi(G)$, where π is a uniformly distributed permutation of $[n]$. Note that if G violates the promise then so does $\pi(G)$, whereas if G satisfies the promise then the probability that D errs on input $\pi(G)$ is at most 2^{-k}. It follows that D errs on input X_n with probability at most 2^{-k}. ■

Extensions. Theorem 2.1 can be extended in various ways. We first note that most natural testers (not only in the setting of testing graph properties in the adjacency matrix model) are "kind of canonical" in the sense that they select some random set of "pivots" and consider small sets of bit-locations as determined by these pivots. That is, randomization is only used in these testers for the selection of the pivots, which induce queries that are each uniformly distributed. Thus, the strategy of the proof of Theorem 2.1 can be applied, resulting in a deterministic algorithm that uses a fixed set of pivots and errs with probability at most 2^{-k} on any input distribution that is invariant under permutations that correspond to mapping among sets of pivots. To formalize the above discussion, we need some definitions.

We turn back to properties of n-bit strings, which we actually view as functions from $[n]$ to $\{0,1\}$. More generally, we shall consider properties of functions from $[n]$ to an arbitrary alphabet Σ. For any set (or rather group) Π of permutations over $[n]$, we say that the property P_n (of such functions) is Π-invariant if for every $f : [n] \to \Sigma$ and every $\pi \in \Pi$ it holds that $f \in \mathsf{P}_n$ if and only $(f \circ \pi) \in \mathsf{P}_n$, where $(f \circ \pi)(i) = f(\pi(i))$ (for every $i \in [n]$). In the following definition, "normality" amounts to non-adaptivity augmented by the requirement that the final decision is deterministic and only depends on the oracle answers, whereas "Π-normality" corresponds to the mapping between the aforementioned pivots.

Definition 2.2 (normal testers): *Let Π be a permutation group over $[n]$ and P_n be a Π-invariant property. We say that a tester for P_n is* normal *if there exists a query-generating algorithm Q and a verdict predicate V such that on internal coins $s \in \{0,1\}^r$ and oracle access to any $f : [n] \to \Sigma$ the tester accepts if and only if $V(f(i_1), ..., f(i_q)) = 1$, where $(i_1, ..., i_q) = Q(s)$. That is, the tester queries the function at locations $i_1, ..., i_q$, which are determined by $Q(s)$ and accepts if and only if the predicate V evaluates to 1 on the q-tuple of answers. We say that the tester is* Π-normal *if the following two conditions hold.*

1. *For every $s, s' \in \{0,1\}^r$ there exists $\pi \in \Pi$ such that $Q(s') = \pi(Q(s))$, where $\pi(i_1, ..., i_q) = (\pi(i_1), ..., \pi(i_q))$.*
2. *For every $s \in \{0,1\}^r$ and $\pi \in \Pi$ there exists $s' \in \{0,1\}^r$ such that $Q(s') = \pi(Q(s))$.*

Note that, by definition, a normal tester is non-adaptive. Also note that the two conditions for Π-normality are equivalent to requiring that *for every $s \in \{0,1\}^r$ and uniformly distributed $\pi \in \Pi$ it holds that $\pi(Q(s))$ is uniformly distributed in* $\{Q(s') : s' \in \{0,1\}^r\}$ (i.e., for every $s, s' \in \{0,1\}^r$ it holds that $\Pr_{\pi \in \Pi}[Q(s') = \pi(Q(s))] = 2^{-r}$).[6] We mention that, indeed, the canonical graph property testers (as defined in [GT] and used in the proof of Theorem 2.1) are normal. Furthermore, they

[6] Clearly, the latter condition implies the two condition in Definition 2.2. To see that the other direction, prove

are $\Pi^{(\mathrm{vr})}$-normal for the group $\Pi^{(\mathrm{vr})}$ of all vertex-relabeling. Other examples of normal testers are discussed below.

Theorem 2.3 (Theorem 2.1, generalized): *Let Π be a permutation group over $[n]$ and P_n be a Π-invariant property. Let X_n be a distribution over functions from $[n]$ to Σ such that for every such function f and every $\pi \in \Pi$ it holds $\Pr[X_n = f] = \Pr[X_n = f \circ \pi]$. Suppose that the promise problem $(\mathsf{P}_n, \Gamma_\epsilon(\mathsf{P}_n))$ can be decided correctly* (in the worst case) *by a Π-normal tester of query complexity $q(n, \epsilon)$ and error probability at most $1/3$. Then, for every $k < n/O(q(n, \epsilon))$, there exists a* (non-adaptive) *deterministic algorithm that inspects the function value at $O(k \cdot q(n, \epsilon))$ fixed and predetermined positions and is correct on a random X_n with probability at least 2^{-k}.*

A distribution X_n as above is called Π-invariant.

Proof: The deterministic algorithm, denoted D, is obtained by fixing the coins to the query-generating algorithm Q. For example, we may query the input function f at locations $(i_1, ..., i_q) = Q(0^r)$, and accept if and only if $V(f(i_1), ..., V(i_q)) = 1$. (Recall that V represents a fixed predicate.) As in the proof of Theorem 2.1, we actually apply this construction after reducing the error probability of the original tester to 2^{-k}.

To analyze the success probability of D on input X_n, we fix any function f and consider the function distribution $f \circ \pi$, where $\pi \in \Pi$ is uniformly distributed. As in the proof of Theorem 2.1, the distribution of the view of the original tester on input f is identical to distribution of the view of the deterministic algorithm D on the randomized input $f \circ \pi$. We conclude that if $f \in \mathsf{P}_n \cup \Gamma_\epsilon(\mathsf{P}_n)$, then the probability that D errs on the input distribution $f \circ \pi$ is at most 2^{-k}. Again, using the hypothesis that X_n is Π-invariant, we conclude that the probability that D errs on input X_n is at most 2^{-k}. ■

Corollaries. Indeed, Theorem 2.1 follows as a special case of Theorem 2.3 by invoking [GT, Thm. 2] (and referring to the group of vertex-relabeling permutations). Next, we illustrate the applicability of Theorem 2.3 to testing low-degree polynomials (see, e.g., [RS]) and to testing monotone functions (see, e.g., [GGLRS]).

- In the case of low-degree tests (see, e.g., [RS]), for some finite field F, we are given a function $f : F^m \to F$ and wish to test whether it is a low-degree polynomial. The standard test selects uniformly at random a line in F^m, queries some points that reside on fixed locations on this line and accepts if and only if an adequate interpolation condition holds. This tester is clearly normal. Furthermore, this tester is Π-normal, where Π is the group of all full-rank affine transformations of F^m (because such transformations define a transitive operation on the set of all pairs of different points).[7] Thus, Theorem 2.3 can be applied to any distribution of functions that is Π-invariant.

- In the case of testing monotonicity (see, e.g., [GGLRS]), for some ordered set S, we are given a function $f : S^m \to \mathsf{R}$ and wish to test whether it is monotone (i.e., whether $f(\alpha) \leq f(\beta)$ for every $\alpha = (\alpha_1, ..., \alpha_m)$ and $\beta = (\beta_1, ..., \beta_m)$ such that $\alpha_i \leq \beta_i$ for every $i \in [m]$). In the case that $S = \{0, 1\}$, the standard test selects uniformly at random two points in S^m

that for any fixed $s, s', s'' \in \{0,1\}^r$ it holds that $p_{s,s'} = p_{s,s''}$, where $p_{a,b} \stackrel{\mathrm{def}}{=} \Pr_{\pi \in \Pi}[Q(b) = \pi(Q(a))]$. (Hint: let π_0 be a fixed permutation satisfying $Q(s'') = \pi_0(Q(s'))$, then $p_{s,s''} = \Pr_{\pi' \in \Pi}[Q(s'') = (\pi_0 \circ \pi')(Q(s))]$, which equals $\Pr_{\pi' \in \Pi}[Q(s') = \pi'(Q(s))] = p_{s,s'}$.)

[7] Note that our notion of normality is closely related (but not identical) to the notion of linear invariances studied in [KS].

that differ in a single coordinate, queries f on these two points, and accepts if and only if an adequate inequality holds. This tester is clearly normal. Furthermore, monotonicity is Π-invariant for the group Π that consists of all permutations $\pi : S^m \to S^m$ such that $\pi(\alpha_1, ..., \alpha_m) = (\alpha_{\pi'(1)}, ..., \alpha_{\pi'(m)})$ for some permutation $\pi' : [m] \to [m]$. Unfortunately, the foregoing tester is not Π-invariant, because the permutations in Π preserve the Hamming weight of strings in $\{0,1\}^m$.

In order to apply Theorem 2.3, we decouple the foregoing tester into m tests such that the i-th test selects uniformly an m-bit string α of Hamming weight i and queries f on this string and on a random string obtained from α by setting one of its 1-entries to zero. Each of these testers is Π-invariant, and so we may apply an adequate extension of Theorem 2.3 that refers to testing properties by a conjunction of several tests.

We comment that similar ideas can be applied even to non-adaptive testers, which seems essential to settings such as testing properties of bounded-degree graphs in the incidence list model (of [GR1]). For example, note that the testers presented in [GR1, GR2] only employ comparison-based computations; that is, they can described in terms of operations such as `select a random vertex`, `select a random neighbor of a given vertex`, and `test equality of two given vertices`.[8] Thus, the operation of these algorithms is maintained when we relabel the vertices. Consequently, they can be derandomized analogously to the proof of Theorem 2.1, resulting in an algorithm that uses a fixed set of vertices and a fixed set of neighbor indices.

Acknowledgments

I am grateful to Omer Reingold and Ronen Shaltiel for extremely useful and insightful discussions. I am also grateful to Marius Zimand for correcting my initial impression by which [Z] can handle any source of linear min-entropy.

Appendix: Generalization of Theorem 1.1

We first detail the generalization of Theorem 1.1 to properties of sequences over any alphabet Σ. This requires generalizing the definition of Γ_ϵ as follows (for any $\mathtt{P}_n \subseteq \Sigma^n$):

$$\Gamma_\epsilon(\mathtt{P}_n) \stackrel{\text{def}}{=} \{x \in \Sigma^n : \forall z \in \mathtt{P}_n \;\; \Delta(x,z) > \epsilon \cdot n\} \tag{2}$$

where $\Delta(x_1 \cdots x_n, z_1 \cdots z_n) = |\{i : x_i \neq z_i\}|$ denotes the number of position in the sequence on which $x = x_1 \cdots x_n$ and $z = z_1 \cdots z_n$ disagree.

Theorem 1.1, generalized. *For any property $\mathtt{P}_n \subseteq \Sigma^n$, it holds that if $|\mathtt{P}_n| \geq \rho \cdot |\Sigma|^n$ and $\epsilon \geq \sqrt{\frac{8\ln(1/\rho)}{n}}$, then $|\Gamma_\epsilon(\mathtt{P}_n)| \leq \rho \cdot |\Sigma|^n$.*

Proof: The proof of Theorem 1.1 generalizes easily, because the proof of Theorem 7.5.3 in [AS] applies (without any change) also to the general case. For sake of self-containment, we reproduce

[8] Many of these algorithms also use the operation of retrieving all neighbors of a given vertex, which can be emulated by successively selecting a random neighbor for sufficiently many times. We also note that in [GR1, GR2] the incidence-lists are sorted, but this is immaterial to the algorithms. For simplicity, here we refer to unsorted incidence-lists.

the proof of [AS, Thm. 7.5.3]. Indeed, the original text refers to $\Sigma = \{0,1\}$ but it actually holds for any finite Σ (provided that Δ and Γ_ϵ are defined as above).

Fixing any $\mathtt{P}_n \subseteq \Sigma^n$, define $\Delta_{\mathtt{P}_n}(x) = \min_{z \in \mathtt{P}_n}\{\Delta(x,z)\}$, and consider a uniformly distributed $\omega \in \Sigma^n$. Then, the theorem's statement can be reformulated as asserting that *if* $\Pr_\omega[\Delta_{\mathtt{P}_n}(\omega)=0] \geq \rho$ *then* $\Pr_\omega[\Delta_{\mathtt{P}_n}(\omega) > \sqrt{8n\ln(1/\rho)}] \leq \rho$. In order to prove this claim, we introduce a martingale (cf. [AS, Chap. 7]) $\zeta_0, ..., \zeta_n$ such that

$$\zeta_i = \zeta_i(\omega) = |\Sigma|^{-(n-i)} \cdot \sum_{r_{i+1},\ldots,r_n \in \Sigma} \Delta_{\mathtt{P}_n}(\omega_1 \cdots \omega_i r_{i+1} \cdots r_n) \tag{3}$$

where $\omega = \omega_1 \cdots \omega_n$. (Indeed, $\zeta_n(\omega) = \Delta_{\mathtt{P}_n}(\omega)$ and $\zeta_0 = \mathrm{E}_\omega[\zeta_n]$.) Note that actually ζ_i only depends on $\omega_1 \cdots \omega_i$. Indeed, the martingale condition holds (i.e., for every fixed $\omega_1 \cdots \omega_i$, it holds that $\mathrm{E}_{\omega_{i+1}}[\zeta_{i+1}|\zeta_i] = \zeta_i$) and $|\zeta_{i+1} - \zeta_i| \leq 1$ (because $|\Delta_{\mathtt{P}_n}(x) - \Delta_{\mathtt{P}_n}(x')| \leq \Delta(x,x')$). By the Martingale Tail Inequality (cf. [AS, Thm. 7.2.1]) we have

$$\Pr_\omega[\zeta_n < \zeta_0 - \lambda\sqrt{n}] \quad < \quad e^{-\lambda^2/2} \tag{4}$$

$$\Pr_\omega[\zeta_n > \zeta_0 + \lambda\sqrt{n}] \quad < \quad e^{-\lambda^2/2} \tag{5}$$

Setting $\lambda = \sqrt{2\log(1/\rho)}$ (such that $\rho = e^{-\lambda^2/2}$) and contrasting Eq. (4) with $\Pr[\zeta_n = 0] \geq \rho$, we conclude that $\zeta_0 \leq \lambda\sqrt{n}$. Thus, Eq. (5) implies $\Pr[\zeta_n > 2\lambda\sqrt{n}] < \rho$, and the theorem follows. ■

References

[AS] N. Alon and J.H. Spencer. *The Probabilistic Method.* John Wiley & Sons, Inc., 1992. Second edition, 2000.

[EKKRV] F. Ergun, S. Kannan, S.R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-Checkers. In *30th STOC*, pages 259–268, 1998.

[ESY] S. Even, A.L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Inform. and Control*, Vol. 61, pages 159–173, 1984.

[F] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, Vol. 75, pages 97–126, 2001.

[FK] A. Frieze and R. Kanan. Quick approximation to matrices and applications. *Combinatorica*, Vol. 19 (2), pages 175–220, 1999.

[GGLRS] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky. Testing Monotonicity. *Combinatorica*, Vol. 20 (3), pages 301–337, 2000.

[GGR] 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.

[GR1] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, pages 302–343, 2002.

[GR2] O. Goldreich and D. Ron. A sublinear bipartite tester for bounded degree graphs. *Combinatorica*, Vol. 19 (3), pages 335–373, 1999.

[GS] O. Goldreich and O. Sheffet. On the randomness complexity of property testing. In *Proc. of RANDOM'07*, to appear.

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

[GW] O. Goldreich and A. Wigderson, Derandomization that is rarely wrong from short advice that is typically good. In the proceedings of *RANDOM'02*, Springer LNCS, Vol. 2483, pages 209–223, 2002.

[KS] T. Kaufman and M. Sudan. Algebraic Property Testing. Unpublished manuscript, 2007.

[R] D. Ron. Property testing. In *Handbook on Randomization, Volume II*, pages 597–649, 2001. (Editors: S. Rajasekaran, P.M. Pardalos, J.H. Reif and J.D.P. Rolim.)

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

[Z] M. Zimand. On derandomizing probabilistic sublinear-time algorithms. In the *Proc. of the 22nd IEEE Conference on Computational Complexity*, pages 1–9, 2007.