Quantum Property Testing[‡]

Harry Buhrman[§] Lance Fortnow[¶] Ilan Newman[‖] Hein Röhrig[**]

January 15, 2006

Abstract

A language L has a property tester if there exists a probabilistic algorithm that given an input x only asks a small number of bits of x and distinguishes the cases as to whether x is in L and x has large Hamming distance from all y in L. We define a similar notion of quantum property testing and show that there exist languages with good quantum property testers but no good classical testers. We also show there exist languages which require a large number of queries even for quantumly testing.

1 Introduction

Suppose we have a large data set, for example, a large chunk of the world-wide web or a genomic sequence. We would like to test whether the data has a certain property, but we may not have the time to look at the entire data set or even a large portion of it.

To handle these types of problems, Rubinfeld and Sudan [36] and Goldreich, Goldwasser and Ron [26] have developed the notion of property testing. Testable properties come in many varieties including graph properties, e.g., [26, 4, 21, 23, 1, 28, 34, 27, 6, 9, 7, 8], algebraic properties of functions [14, 36, 19], Boolean functions and languages [5, 22] and geometric objects [3, 17]. Nice surveys of this area can be found in [35, 20].

In this model, the property tester has random access to the n input bits similar to the black-box oracle model. The tester can query only a small number of input bits; the set of indices is usually of constant size and chosen probabilistically. Clearly we cannot determine from this small number of bits whether the input sits in some language L. However, for many languages we can distinguish the case that the input is in L from the case that the input differs from all inputs in L of the same length by some constant fraction of input bits. Note that we do not consider the time complexity of the testing algorithms in this paper.

Since there are many examples where quantum computation gives us an advantage over classical computation [13, 38, 37, 29] one may naturally ask whether using quantum computation may lead to better property testers. By using the quantum oracle-query model developed by Beals, Buhrman, Cleve, Mosca, and de Wolf [11] we can easily extend the definitions of property testing to the quantum setting.

[‡]Research done while all authors were visiting the NEC Research Institute. A preliminary version of this paper appeared in SODA 2003 [16].

[§]CWI and University of Amsterdam; partially supported by the EU fifth framework project QAIP, IST-1999-11234.

[¶]NEC Research Institute. Current affiliation: University of Chicago

[‖]Haifa University, partially supported by the Israeli Science Foundation (ISF) 55/03

[**]CWI; partially supported by the EU fifth framework project QAIP, IST-1999-11234. Current affiliation: Google, Inc.

Beals et al. [11] have shown that for all total functions we have a polynomial relationship between the number of queries required by quantum machine and that needed by a classical machine. For greater separations one needs to impose a promise on the input. The known examples, such as those due to Simon [38] and Bernstein and Vazirani [13], require considerable structure in the promise. Property testing amounts to the natural promise of either being in the language or far from each input in the language. This promise would seem to have too little structure to give a separation but in fact we can prove that quantum property testing can greatly improve on classical testing.

We show that every subset of Hadamard codes has a quantum property tester with $\mathrm{O}(1)$ queries and that most subsets would require $\Theta(\log n)$ queries to test with a probabilistic tester. This shows that indeed quantum property testers are more powerful than classical testers. Moreover, we also give an example of a language where the quantum tester is exponentially more efficient.

Beals et al. observed that every k-query quantum algorithm gives rise to a degree-$2k$ polynomial in the input bits, which gives the acceptance probability of the algorithm; thus, a quantum property tester for P gives rise to a polynomial that is on all binary inputs between 0 and 1, that is at least $2/3$ on inputs with the property P and at most $1/3$ on inputs far from having the property P. Szegedy [40] asked whether it is possible to algebraically characterize the complexity of classical testing by the minimum degree of such polynomials; however, our separation results imply that there are properties, for which such polynomials have constant degree, but for which the best classical tester needs $\Omega(\log n)$ queries. Hence, the minimum degree is only a lower bound, which sometimes is not tight.

A priori it is conceivable that every language has a quantum property tester with a small number of queries. We show that this is not the case. We prove that for most properties of a certain size, every quantum algorithm requires $\Omega(n)$ queries. We then show that a natural property, namely, the range of a d-wise independent pseudorandom generator cannot be quantumly tested with less than $(d+1)/2$ queries for every odd $d \leq n/\log n - 1$.

While our paper is the first to explicitly consider property testing in the quantum setting, several previous papers have considered related testing problems [32, 18]. The algorithms of Hales and Hallgren [30] give a property tester for periodicity when the bad function is also periodic.

2 Preliminaries

We will use the following formal definition of property testing from Goldreich [25]:

Definition 1 *Let S be a finite set, and P a set of functions mapping S to $\{0,1\}$. A* property tester *for P is a probabilistic oracle machine M, which given a distance parameter $\varepsilon > 0$ and oracle access to a function $f : S \to \{0,1\}$, satisfies the following conditions:*

1. *the tester accepts f if it is in P: if $f \in P$ then $\Pr(M^f(\varepsilon) = 1) \geq 2/3$*
2. *the tester rejects f if it is far from P: if $|\{x \in S : f(x) \neq g(x)\}| > \varepsilon \cdot |S|$, for every $g \in P$, then $\Pr(M^f(\varepsilon) = 1) \leq 1/3$.*

Here M^f denotes that the machine M is provided with the oracle for f.

Definition 2 *The complexity of the tester is the number of oracle queries it makes: a property P has an (ε, q)-*tester *if there is a tester for P that makes at most q oracle queries for distance parameter ε.*

We often consider a language $L \subseteq \{0,1\}^*$ as the family of properties $\{P_n\}$ with P_n the characteristic functions of the length-n strings from L, and analyze the query complexity $q = q(\varepsilon, n)$ asymptotically for large n.

To define quantum property testing we simply modify Definition 1 by allowing M to be a quantum oracle machine. We need to be careful to make sure our oracle queries are unitary operations. If $|f(x)| = |g(y)|$ for all $x, y \in S$ and $f, g \in P$, we use the oracle-query model by Beals, Buhrman, Cleve, Mosca and de Wolf [11]: we define the unitary transformation U_f that maps the basis state $|x, y, z\rangle$ to $|x, y \oplus f(x), z\rangle$ where $|x| = \lceil \log |S| \rceil$, $|y| = |f(x)|$ and $\oplus$ denotes bitwise exclusive or. In case there are x, y, f, g so that $|f(x)| \neq |g(y)|$, we define U_f as mapping $|x, l, y, z\rangle$ to $\left|x, l + |f(x)| \mod k, y \oplus 0^{k-|f(x)|} f(x), z\right\rangle$ where $k = \max\{|f(x)| : f \in P \text{ and } x \in S\}$, $|x| = \lceil \log |S| \rceil$, $|l| = \lceil \log k \rceil$, and $|y| = k$.

We recommend the book of Nielsen and Chuang [33] for background information on quantum computing.

3 Separating Quantum and Classical Property Testing

We show that there exist languages with $(\varepsilon, \mathrm{O}(1))$ quantum property testers that do not have $(\varepsilon, \mathrm{O}(1))$ classical testers.

Theorem 1 *There is a language L that is ε-testable by a quantum test with $\mathrm{O}(1/\varepsilon)$ queries but for which every probabilistic ε-test with $\varepsilon \leq 1/2$ requires $\Omega(\log n)$ queries.*

We use Hadamard codes to provide examples for Theorem 1:

Definition 3 *The* Hadamard code *of $y \in \{0,1\}^{\log n}$ is $x = h(y) \in \{0,1\}^n$, where $x_i = y \cdot i$ for every index i. Here $i \in \{0, \ldots, n-1\}$ and we identify $\{0, \ldots, n-1\}$ with $\{0,1\}^{\log n}$; $y \cdot i = \sum_{j=0}^{\log n - 1} y_j i_j \mod 2$ denotes the inner product of two vectors $y, i \in \mathbb{F}_2^{\log n}$.*

Note: the Hadamard mapping $h : \{0,1\}^{\log n} \to \{0,1\}^n$ is one-to-one. Bernstein and Vazirani [13] showed that given a codeword x, a quantum computer can extract the y for which $x = h(y)$ with one query to an oracle for the bits of x, whereas a classical probabilistic procedure needs $\Omega(\log n)$ queries. Based on this separation for a decision problem we construct for $A \subseteq \{0,1\}^{\log n}$ the property $P_A \subseteq \{0,1\}^n$,

$$P_A := \{x : \exists y \in A \text{ s.t. } x = h(y)\}.$$

Theorem 1 follows from the following two lemmas.

Lemma 2 *For every A, P_A has an $(\varepsilon, \mathrm{O}(1/\varepsilon))$ quantum tester. Furthermore, the test has one-sided error.*

Lemma 3 *For most $A \subseteq \{0,1\}^{\log n}$, P_A requires $\Omega(\log n)$ queries for a 1/2-test, even for testers with two-sided error.*

Before we prove Lemma 2 we note that for every A, P_A can be tested by a classical one-sided-error algorithm with $\mathrm{O}(1/\varepsilon + \log n)$ queries even nonadaptively; hence, the result of Lemma 3 is tight. An $\mathrm{O}(1/\varepsilon \log n)$-test follows from Theorem 4 below. The slightly more efficient test, of query complexity $\log n + \mathrm{O}(1/\varepsilon)$, is the following: First we query x_{2^i}, $i = 0, \ldots, \log n - 1$. Note that if $x = h(y)$ then $y_i = x_{2^i}$ for $i = 0, \ldots, \log n - 1$. Thus a candidate y for $x = h(y)$ is found. If $y \notin A$ then x is rejected. Then $k = \mathrm{O}(1/\varepsilon)$ times the following check is performed: a index $i \in \{0, \ldots, n-1\}$ is chosen independently and uniformly at random and if $x_i \neq y \cdot i$, then x is rejected. Otherwise, x is accepted. Clearly if x is rejected then $x \notin P_A$. It is easily verified that if x has Hamming distance more than εn from every z in P_A then with constant probability x is rejected.

Proof of Lemma 2. P_A can be checked with $\mathrm{O}(1/\varepsilon)$ queries on a quantum computer: The test is similar to the test above except that y can be found in $\mathrm{O}(1)$ queries: k times query for random i, j values x_i, x_j, and $x_{i\oplus j}$. If $x_i \oplus x_j \neq x_{i\oplus j}$ reject. $k = \mathrm{O}(1/\varepsilon)$ is sufficient to detect an input x that is εn-far from being a Hadamard codeword with high probability. Now run the Bernstein-Vazirani algorithm to obtain y. Accept if and only if $y \in A$. Obviously, if $x \in P_A$, the given procedure accepts, and if x is far from each $x' \in P_A$, then it is either far from being a Hadamard codeword or it is close to a Hadamard codeword $h(y')$ for a $y' \notin A$; note that in this case x is far from every $h(y)$, $y \in A$ as two distinct Hadamard codewords are of Hamming distance $n/2$. Thus, in this case the second part of the tester succeeds with high probability in finding y' and rejects because $y' \notin A$. We note also that this algorithm has one-sided error. □

Proof of Lemma 3. The lower bound makes use of the Yao principle [41]: let D be an arbitrary probability distribution on positive and negative inputs, i.e., on inputs that either belong to P_A or are $n/2$-far from P_A. Then if every deterministic algorithm that makes at most q queries, errs with probability at least $1/16$ with respect to input chosen according to D, then q is a lower bound on the number of queries of any randomized algorithm for testing P_A with error probability bounded by $1/16$.

For our lower bound, D will be the uniform distribution over Hadamard codewords of length n, namely, generated by choosing $y \in \{0,1\}^{\log n}$ uniformly at random and setting $x = h(y)$. Note that for any $A \subset \{0,1\}^{\log n}$, D is concentrated on positive and negative inputs as required, as two Hadamard codewords are of Hamming distance $n/2$ apart.

The lower bound will be established by a counting argument. We show that for a fixed tester that makes $q \leq (\log n)/2$ queries, the probability over random choices of A that the algorithm errs on at most $1/16$ of the inputs is much less than $1/T$, where T is the number of such algorithms. By the union bound it follows that for most properties there is no such algorithm.

Indeed, choose $A \subseteq \{0,1\}^{\log n}$ by picking independently each $i \in \{0,1\}^{\log n}$ to be in A with probability $1/2$. Let $\mathcal{T}$ be any fixed deterministic decision tree performing at most q queries in every branch. Let $A_{\mathcal{T}} := \{y | \mathcal{T}(h(y)) = \mathrm{accept}\}$ and let $\mathrm{err}(\mathcal{T}, A) := \big|(A \setminus A_{\mathcal{T}}) \cup (A_{\mathcal{T}} \setminus A)\big|/n$ denote the error probability of $\mathcal{T}$ for property P_A with input distribution D. Assume first that $|A_{\mathcal{T}}| \leq n/2$. Since for a $h(y) \in \{0,1\}^n$ chosen according to D we have $\Pr_y[\mathcal{T}(h(y)) = \mathrm{accept}] = |A_{\mathcal{T}}|/n \leq 1/2$, it follows by a Chernoff-type bound [10] that $\Pr_A[|A \cap A_{\mathcal{T}}| \geq 3/4|A|] \leq 2^{-n/8}$. If $|A \cap A_{\mathcal{T}}| < 3/4|A|$ then $\mathcal{T}$ will be wrong on at least $|A|/4$ of the positive inputs. With high probability, A is not too small: a Chernoff-type bound implies $\Pr_A[|A| \leq n/4] \leq 2^{-n/16}$. Then $\Pr_A[\mathrm{err}(\mathcal{T}, A) < 1/16] \leq \Pr_A\big[|A| \leq n/4\big] + \Pr_A\big[\mathrm{err}(\mathcal{T}, A) < 1/16 \;\big|\; |A| \geq n/4\big] \leq 2^{-n/8} + 2^{-n/16} \leq 2 \cdot 2^{-n/16}$. If $|A_{\mathcal{T}}| > n/2$ the same reasoning shows that with probability of at most $2 \cdot 2^{-n/16}$, $\mathcal{T}$ will err with D-probability less than $1/16$ on the negative inputs. Overall, we have for every fixed $\mathcal{T}$

$$\Pr_A[\mathrm{err}(\mathcal{T}, A) \leq 1/16] \leq 2 \cdot 2^{-n/16} \quad .$$

Now, let us bound from above the number T of algorithms that make at most q queries. As an algorithm may be adaptive, it can be defined by $2^q - 1$ query positions for all queries on all branches and a Boolean function $f : \{0,1\}^q \to \{\mathrm{accept}, \mathrm{reject}\}$ of the decision made by the algorithm for the possible answers. Hence, there are at most $T \leq (2n)^{2^q}$ such algorithms. However, for $q \leq (\log n)/2$, we have $T \cdot 2 \cdot 2^{-n/16} = \mathrm{o}(1)$, which shows that for most A as above, every $1/2$-test that queries at most $(\log n)/2$ many queries has error probability of at least $1/16$. Standard amplification techniques then imply that for some constant c, every $1/2$-test that performs $c \log n$ many queries has error greater than $1/3$. □

Theorem 4 *Let $P \subseteq \{0,1\}^n$ be a property with $|P| > 0$. For any $\varepsilon > 0$, P can be ε-tested by a one-sided error classical algorithm using $\mathrm{O}((\log |P|)/\varepsilon)$ queries.*

Proof. Denote the input by $y \in \{0,1\}^n$ and $s := |P|$. Consider the following algorithm: query the input y in $k := \ln(3s^2)/\varepsilon$ random places; accept if there is at least one $x \in P$ consistent with the bits from the input and reject otherwise. Clearly, if $y \in P$, this algorithm works correctly.

If y is ε-far from each $x \in P$, then for every specific $x \in P$, $\Pr[x_i = y_i] \leq 1 - \varepsilon$ when choosing an $i \in [n]$ uniformly at random. With k indices chosen independently and uniformly at random, the probability for no disagreement with x becomes $(1-\varepsilon)^k \leq 1/(3s^2)$. Therefore, the probability that there is no disagreement for at least one of the s members of P is at most $1/(3s)$, so with probability $2/3$ for a y that is far from P, we will rule out every $x \in P$ as being consistent with y. $\square$

4 An Exponential Separation

In this section, we show that a quantum computer can be exponentially more efficient than a classical computer in testing certain properties.

Theorem 5 *There exists a language L that for every $\varepsilon = \Omega(1)$ is $(\varepsilon,\ \log n\ \log\log n)$ quantumly testable but every classical $1/8$-test for L requires $\Omega(\sqrt{n})$ queries.*

The language that we provide is inspired by Simon's problem [38] and our quantum testing algorithm makes use of Brassard and Høyer's algorithm for Simon's problem [15]. Simon's problem is to find $s \in \{0,1\}^n \setminus \{0^n\}$ from a function-query oracle for some $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that $f(x) = f(y) \Leftrightarrow x = y \oplus s$. Simon proved that classically, $\Omega(2^{n/2})$ queries are required on average to find s, and gave a quantum algorithm for determining s with an expected number of queries that is polynomial in n; Brassard and Høyer improved the quantum algorithm to worst-case polynomial time. Their algorithm produces in each run a z with $z \cdot s = 0$ that is linearly independent to all previously computed such zs. Essentially, our quantum tester uses this subroutine to try to extract information about s until it fails repeatedly. Høyer [31] analyzed this approach in group-theoretic terms, obtaining an alternative proof to Theorem 7. Friedl et al. [24] generalize Theorem 7 to hold for languages based on any finite Abelian group.

In the following, let $N = 2^n$ denote the length of the binary string encoding a function $f : \{0,1\}^n \to \{0,1\}$. For $x \in \{0,1\}^n$ let $x[j]$ denote the jth bit of x, i.e., $x = x[1] \ldots x[n]$. We define

$$L := \{f \in \{0,1\}^N : \exists s \in \{0,1\}^n \setminus \{0^n\}\ \forall x \in \{0,1\}^n\ f(x) = f(x \oplus s)\}$$

Theorem 5 follows from the following two theorems.

Theorem 6 *Every classical $1/8$-tester for L must make $\Omega(\sqrt{N})$ queries, even when allowing two-sided error.*

Theorem 7 *There is a quantum property tester for L making $\mathrm{O}(\log N\ \log\log N)$ queries. Moreover, this quantum property tester makes all its queries nonadaptively.*

Proof of Theorem 6. We again apply the Yao principle [41] as in the proof of Lemma 3: we construct two distributions, P and U, on positive and at least $(N/8)$-far negative inputs, respectively, and let D be a distribution that is defined by $D = (P+U)/2$. We will show that every adaptive decision tree $\mathcal{T}$ has error $1/2 - \mathrm{o}(1)$ on a random input chosen according to D.

The distribution P is defined as follows: We first choose $s \in \{0,1\}^n$ at random. This defines a matching M_s of $\{0,1\}^n$ by matching x with $x \oplus s$. Now a function f_s is defined by choosing for each matched pair independently $f_s(x) = f_s(x \oplus s) = 1$ with probability $1/2$ and $f_s(x) = f_s(x \oplus s) = 0$ with probability $1/2$. Clearly, this defines a distribution that is concentrated on positive inputs. Note that it might be that by choosing different ss we end up choosing the same function. However, this will be considered different events in the probability space, i.e., the atomic events in P are the pairs (s, f_s) as described above.

Now let $\tilde{U}$ be the uniform distribution over all functions: we select the function by choosing for each x independently $f(x) = 1$ with probability $1/2$ and 0 with probability $1/2$. Since every function has a nonzero probability, $\tilde{U}$ is not supported exclusively on the negative instances. We define U to be $\tilde{U}$ conditioned on the event that the input is $N/8$ far from the property. As we show in Lemma 8, a function chosen according to $\tilde{U}$ is $N/8$ far from having the property with very high probability, hence it will be a good approximation for U.

Let $\mathcal{T}$ be any deterministic decision tree. Let v be a vertex of $\mathcal{T}$. We will show that for every vertex v of small depth in $\mathcal{T}$, $\Pr_P[\text{input } f \text{ is consistent with } v] = \Pr_U[\text{input } f \text{ is consistent with } v](1+\mathrm{o}(1))$, from which we will conclude that $\mathcal{T}$ has error $1/2 - \mathrm{o}(1)$.

Definition 4 *For $f : \{0,1\}^n \to \{0,1\}$ and $s \in \{0,1\}^n$ we define $n_s := |\{x : f(x) = f(x \oplus s)\}|$.*

Lemma 8 *Let f be chosen according to $\tilde{U}$. Then $\Pr_{\tilde{U}}[\exists s \in \{0,1\}^n : n_s \geq 3N/4] = e^{-\Omega(N)}$.*

Proof. Let f be chosen according to $\tilde{U}$ and $s \in \{0,1\}^n$. By a Chernoff bound [10], we obtain $\Pr_{\tilde{U}}[n_s \geq 3N/4] = e^{-\Omega(N)}$ for every fixed s. Together with the union bound over all $2^n = N$ choices of s this yields $\Pr_{\tilde{U}}[\exists s \in \{0,1\}^n : n_s \geq 3N/4] = N \cdot e^{-\Omega(N)} = e^{-\Omega(N)}$. □

For every s, we need to change $(N - n_s)/2$ values of f to get an input f' that has the property $f'(x) = f'(x \oplus s)$ for all x. Hence, Lemma 8 implies that with probability $1 - e^{-\Omega(N)}$ an input chosen according to $\tilde{U}$ will be $N/8$ far from having the property.

From the definition of $\tilde{U}$, we immediately obtain the following:

Lemma 9 *Let $\mathcal{T}$ be any fixed deterministic decision tree and let v be a vertex of depth d in $\mathcal{T}$. Then $\Pr_{\tilde{U}}[f$ is consistent with the path to $v] = 2^{-d}$.*

We now want to derive a similar bound as in Lemma 9 for functions chosen according to P. For this we need the following definition for the event that after d queries, nothing has been learned about the hidden s:

Definition 5 *Let $\mathcal{T}$ be a deterministic decision tree and u a vertex in $\mathcal{T}$ at depth d. We denote the path from the root of $\mathcal{T}$ to v by $\mathrm{path}(v)$. Every vertex v in $\mathcal{T}$ defines a query position $x_v \in \{0,1\}^n$. For $f = f_s$ chosen according to P, we denote by B_v the event $B_v := \{(s, f_s) : s \neq x_u \oplus x_w$ for all $u, w \in \mathrm{path}(v)\}$.*

Lemma 10 *Let v be a vertex of depth d in a decision tree $\mathcal{T}$. Then $\Pr_P[B_v] \geq 1 - \binom{d-1}{2}/N$*

Proof. B_v does not occur if for some v, w on the path to v we have $s = x_v \oplus x_w$. As there are $d - 1$ such vertices, there are at most $\binom{d-1}{2}$ pairs. Each of these pairs excludes exactly one s and there are N possible values of s. □

Lemma 11 *Let v be a vertex of depth d in a decision tree $\mathcal{T}$ and let f be chosen according to P. Then $\Pr_P[f$ is consistent with $v | B_v] = 2^{-d}$.*

Proof. By the definition of P, f gets independently random values on vertices that are not matched. But if B_v occurs, then no two vertices along the path to v are matched and hence the claim follows. $\square$

Now we can complete the proof of the theorem: assume that $\mathcal{T}$ is a deterministic decision tree of depth $d = \mathrm{o}(\sqrt{N})$ and let v be any leaf of $\mathcal{T}$. Then by Lemmas 10 and 11, we get that $\Pr_P[f \text{ is consistent with } v] = (1 \pm \mathrm{o}(1))2^{-d}$. On the other hand, by Lemmas 8 and 9 we get that $\Pr_U[f \text{ is consistent with } v] = (1 \pm \mathrm{o}(1))2^{-d}$ and hence $\mathcal{T}$ has only $\mathrm{o}(1)$ bias factor of being right on every leaf. This implies that its error probability is $1/2 - \mathrm{o}(1)$. $\square$

Proof of Theorem 7. We give a quantum algorithm making $\mathrm{O}(\log N \log\log N)$ queries to the quantum oracle for input $f \in \{0,1\}^N$. We will show that it accepts with probability 1 if $f \in L$ and rejects with high probability if the Hamming distance between f and every $g \in L$ is at least εN. Pseudo code for our algorithm is given on page 8; it consists of a classical main program SimonTester and a quantum subroutine SimonSampler adapted from Brassard and Høyer's algorithm for Simon's problem [15, Section 4]. The quantum gates used are the 2^n-dimensional Hadamard transform H_{2^n}, which applies

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

individually to each of n qubits, the quantum oracle query U_f, and classical reversible operations run in quantum superposition.

The following technical lemma captures the operation of the quantum subroutine SimonSampler. For $i_1, \dots, i_J$ fixed, let $Y_J := \{y \in \{0,1\}^n : \forall j \leq J \; y[i_j] = 0\}$ denote the length-n binary strings that are 0 at positions $i_1, \dots, i_J$.

Lemma 12 *When* SimonSampler *is passed k linearly independent vectors $z_1, \dots, z_k$ so that all $i_j := \min\{i : z_j[i] = 1\}$ are distinct for $1 \leq j \leq k$, then the state $|\psi\rangle$ before the measurement is*

$$\frac{\sqrt{2^k}}{N} \sum_{x \in \{0,1\}^n} \sum_{y \in Y_k} (-1)^{x \cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_k\rangle \;.$$

Proof. We follow the steps of subroutine SimonSampler when passed k linearly independent vectors $z_1, \dots, z_k$ so that all $i_j := \min\{i : z_j[i] = 1\}$ are distinct for $1 \leq j \leq k$.

$$\begin{aligned} |0^n\rangle|0\rangle|0^k\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle|0\rangle|0^k\rangle &\mapsto \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle|0^k\rangle \\ &\mapsto \frac{1}{N} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle|f(x)\rangle|0^k\rangle \end{aligned}$$

This is the state before the **for** loop is entered. We claim and proceed to show by induction that after the Jth execution of the loop body, the state is

$$\frac{\sqrt{2^J}}{N} \sum_{x \in \{0,1\}^n} \sum_{y \in Y_J} (-1)^{x \cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle |0^{k-J}\rangle \;.$$

Executing the body of the loop for $j = J + 1$,

$$\frac{\sqrt{2^J}}{N} \sum_{x \in \{0,1\}^n} \sum_{y \in Y_J} (-1)^{x \cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle |0\rangle |0^{k-J-1}\rangle$$

Procedure SimonTester

```
for k = 0 to n − 1 do
    l ← 0
    repeat
        z ← SimonSampler(z_1, ..., z_k)
        l ← l + 1
    until z ≠ 0^n or l > 2(log n)/ε^2
    if z = 0^n then
        accept
    else
        z_{k+1} ← z
reject
```

Procedure SimonSampler$(z_1, \ldots, z_k)$

```
input: z_1, ..., z_k ∈ {0,1}^n
output: z ∈ {0,1}^n
quantum workspace: X ⊗ Y ⊗ Z where
X is n qubits X = X_1 ⊗ ··· ⊗ X_n, X_i = C^2,
Y = C^2 is one qubit, and
Z is k qubits Z = Z_1 ⊗ ··· ⊗ Z_k, Z_j = C^2
initialize the workspace to |0^n⟩|0⟩|0^k⟩
apply H_{2^n} to X
apply U_f to X ⊗ Y
apply H_{2^n} to X
for j = 1 to k do
    i ← min{i : z_j[i] = 1}
    apply CNOT with control X_i and target Z_j
    apply |x⟩ ↦ |x ⊕ z_j⟩ to X conditional on Z_j
    apply H_2 to Z_j
return measurement of X
```

$$\mapsto \frac{\sqrt{2^J}}{N} \sum_{x\in\{0,1\}^n} \sum_{y\in Y_J} (-1)^{x\cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle |y[i_{j+1}]\rangle |0^{k-J-1}\rangle$$

$$= \frac{\sqrt{2^J}}{N} \sum_{\substack{x\in\{0,1\}^n \\ y\in Y_{J+1} \\ b\in\{0,1\}}} (-1)^{x\cdot(y\oplus bz_{J+1})} |y \oplus bz_{J+1}\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle |b\rangle |0^{k-J-1}\rangle$$

(Here, we used the fact that $Y_J = Y_{J+1} \dot{\cup} (z_{J+1} \oplus Y_{J+1})$.)

$$\mapsto \frac{\sqrt{2^J}}{N} \sum_{\substack{x\in\{0,1\}^n \\ y\in Y_{J+1} \\ b\in\{0,1\}}} (-1)^{x\cdot(y\oplus bz_{J+1})} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle |b\rangle |0^{k-J-1}\rangle$$

$$= \frac{\sqrt{2^{J+1}}}{N} \sum_{x\in\{0,1\}^n} \sum_{y\in Y_{J+1}} (-1)^{x\cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_J\rangle \frac{1}{\sqrt{2}} \sum_{b\in\{0,1\}} (-1)^{x\cdot(bz_{J+1})} |b\rangle |0^{k-J-1}\rangle$$

$$\mapsto \frac{\sqrt{2^{J+1}}}{N} \sum_{x\in\{0,1\}^n} \sum_{y\in Y_{J+1}} (-1)^{x\cdot y} |y\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_{J+1}\rangle |0^{k-J-1}\rangle \ .$$

□

As an immediate consequence, we can establish the invariant that in SimonTester $\{z_1, \ldots, z_k\}$ is always linearly independent with $i_j = \min\{i : z_j[i] = 1\}$ distinct for $1 \le j \le k$; moreover, if $f \in L$, then just as in Simon's algorithm, a nonzero z is orthogonal to the hidden s:

Lemma 13 *If measuring the first register, $\mathcal{X}$, yields a nonzero value z, then*

1. *$\{z_1, \ldots, z_k, z\}$ is linearly independent,*
2. *$\min\{i : z[i] = 1\}$ is distinct from i_j for $1 \le j \le k$, and*
3. *if $f \in L$, then $z \cdot s = 0$ for every $s \ne 0^n$ such that $f(x) = f(x \oplus s)$ for all x.*

Proof. If we measure the state from Lemma 12, then for the value z of the first register holds $z \in Y_k$. This implies 2, from which follows 1. For 3: as in Simon's original algorithm, if there is a $s \ne 0^n$ so that for all x, $f(x) = f(x \oplus s)$, then we can rewrite the state from Lemma 12 as

$$\frac{\sqrt{2^k}}{N} \sum_{\substack{x:x<x\oplus s \\ y\in Y_k}} |y\rangle \Big((-1)^{x\cdot y} |f(x)\rangle + (-1)^{(x\oplus s)\cdot y} |f(x \oplus s)\rangle \Big) |x \cdot z_1\rangle \cdots |x \cdot z_k\rangle$$

$$= \frac{\sqrt{2^k}}{N} \sum_{x:x<x\oplus s} \sum_{y\in Y_k} |y\rangle (-1)^{x\cdot y} \left(1 + (-1)^{s\cdot y}\right) |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_k\rangle \ .$$

Hence, only y with $s \cdot y = 0$ will have nonzero amplitude. □

Next, we want to assess the probability of obtaining $z = 0^n$ in SimonTester Line 4. We let P_0 denote the projection operator mapping $|0^n\rangle|y\rangle|z\rangle \mapsto |0^n\rangle|y\rangle|z\rangle$ and $|x\rangle|y\rangle|z\rangle \mapsto 0$ for $x \ne 0^n$; hence, $\|P_0|\psi\rangle\|^2$ is the probability of obtaining 0 when measuring subspace $\mathcal{X}$ of the quantum register in state $|\psi\rangle$. We can characterize the probability for outcome $z = 0^n$ in terms of the following definition and lemma:

Definition 6 *For $c \in \{0,1\}^k$ and $z_1, \ldots, z_k \in \{0,1\}^n$ we define $D_c := \{x \in \{0,1\}^n : x \cdot z_1 = c[1], \ldots, x \cdot z_k = c[k]\}$.*

Lemma 14 *Let $|\psi\rangle$ be the state before the measurement in* SimonSampler*, when* SimonSampler *is passed k linearly independent vectors $z_1, \ldots, z_k$ so that all $i_j := \min\{i : z_j[i] = 1\}$ are distinct for $1 \le j \le k$.*

1. *$\|P_0|\psi\rangle\|^2 = 1$ if and only if for every $c \in \{0,1\}^k$, f is constant when restricted to D_c.*
2. *If $\|P_0|\psi\rangle\|^2 \ge 1 - \varepsilon^2/2$, then f differs in at most εN points from some function g that is constant when restricted to D_c for every $c \in \{0,1\}^k$.*

Proof. For $b \in \{0,1\}$ let $D_{b,c} := D_c \cap f^{-1}\{b\} = \{x : f(x) = b \text{ and } x \cdot z_1 = c[1], \ldots, x \cdot z_k = c[k]\}$. Note that the $D_{b,c}$ and D_c also depend on $z_1, \ldots, z_k$ and the $D_{b,c}$ depend on f. Let

$$\begin{aligned} |\psi_0\rangle &:= \frac{\sqrt{2^k}}{N} \sum_{x \in \{0,1\}^n} |0^n\rangle |f(x)\rangle |x \cdot z_1\rangle \cdots |x \cdot z_k\rangle \\ &= \frac{\sqrt{2^k}}{N} \sum_{b \in \{0,1\}} \sum_{c \in \{0,1\}^k} |D_{b,c}|\, |0^n\rangle |b\rangle |c[1]\rangle \cdots |c[k]\rangle \ . \end{aligned}$$

By Lemma 12, at the end of SimonSampler the system is in state $|\psi\rangle = |\psi_0\rangle + |\psi_0^\perp\rangle$ for some $|\psi_0^\perp\rangle$ orthogonal to $|\psi_0\rangle$. We consider the case $\|P_0|\psi\rangle\|^2 = 1$. Then the register $\mathcal{X}$ must be in state $|0^n\rangle$ and thus $|\psi\rangle = |\psi_0\rangle$. Since the state has norm 1, we know that

$$\sum_{b \in \{0,1\}} \sum_{c \in \{0,1\}^k} |D_{b,c}|^2 = \frac{N^2}{2^k} \ . \tag{1}$$

The $D_{b,c}$ partition $\{0,1\}^n$ and the $D_c = D_{0,c} \cup D_{1,c}$ have the same size for all $c \in \{0,1\}^k$ because they are cosets of D_0. Therefore,

$$\sum_{b \in \{0,1\}} \sum_{c \in \{0,1\}^k} |D_{b,c}| = N \text{ and } |D_{0,c}| + |D_{1,c}| = \frac{N}{2^k} \text{ for all } c \in \{0,1\}^k \ . \tag{2}$$

$|D_{0,c}|^2 + |D_{1,c}|^2 \le N^2/2^{2k}$, but in order for equation (1) to hold, $|D_{0,c}|^2 + |D_{1,c}|^2$ must be exactly $N^2/2^{2k}$. This can only be achieved if either $D_{0,c}$ or $D_{1,c}$ is empty. So f must be constant when restricted to D_c for any $c \in \{0,1\}^k$. Conversely, if f is constant when restricted to D_c for any $c \in \{0,1\}^k$, then equation (1) holds, therefore $\||\psi_0\rangle\| = 1$ and $|\psi\rangle = |\psi_0\rangle$. This concludes the proof of case 1 of the lemma.

If $\|P_0|\psi\rangle\|^2 = \||\psi_0\rangle\|^2 \ge 1 - \delta$, then

$$\sum_{b \in \{0,1\}} \sum_{c \in \{0,1\}^k} |D_{b,c}|^2 \ge (1 - \delta) \frac{N^2}{2^k} \ . \tag{3}$$

Still, the constraints (2) hold; let $r2^k$ be the number of $c \in \{0,1\}^k$ so that $\min\{|D_{0,c}|, |D_{1,c}|\} \ge \gamma N/2^k$. Then

$$\sum_{b \in \{0,1\}} \sum_{c \in \{0,1\}^k} |D_{b,c}|^2 \le r2^k(\gamma^2 + (1-\gamma)^2)\frac{N^2}{2^{2k}} + (1-r)2^k \frac{N^2}{2^{2k}} \ ,$$

and using (3), we obtain $r \leq \delta/(1-\gamma^2-(1-\gamma)^2)$. With $\delta = \varepsilon^2/2$ and $\gamma = \varepsilon/2$, this implies $r \leq \varepsilon$. But then

$$\sum_{c \in \{0,1\}^k} \min\{|D_{0,c}|, |D_{1,c}|\} \leq r 2^k \frac{N}{2^{k+1}} + (1-r) 2^k \gamma \frac{N}{2^k} \leq \varepsilon N \ .$$

□

We need to relate these two cases to membership in L and bound the number of repetitions needed to distinguish between the two cases. This is achieved by the following two lemmas.

Lemma 15 *Let k be the minimum number of linearly independent vectors $z_1, \ldots, z_k$ so that for each $c \in \{0,1\}^k$, f is constant when restricted to D_c. Then $f \in L$ if and only if $k < n$.*

Proof. If $k < n$, then there exists an $s \neq 0^n$ with $s \cdot z_1 = 0, \ldots, s \cdot z_k = 0$. For each such s and all x, we have $x \cdot z_1 = (x \oplus s) \cdot z_1, \ldots, x \cdot z_k = (x \oplus s) \cdot z_k$ and $x \in D_{f(x), x\cdot z_1, \ldots, x \cdot z_k}$ and $x \oplus s \in D_{f(x \oplus s), x \cdot z_1, \ldots, x \cdot z_k}$, therefore $f(x) = f(x \oplus s)$. Conversely, for $f \in L$, $S := \{s : \forall x f(x) = f(x \oplus s)\}$ is a nontrivial subspace of $\{0,1\}^n$, therefore $S^\perp = \{z : z \cdot s = 0 \forall s \in S\}$ is a proper subspace of $\{0,1\}^n$. Let z_1, $\ldots$, z_k be an arbitrary basis of $S^\perp$. □

Lemma 16 *Let $0 < q < 1$, and $|\varphi_1\rangle, \ldots, |\varphi_m\rangle$ be quantum states satisfying $\|P_0|\varphi_j\rangle\|^2 < 1-\delta$ for $1 \leq j \leq m$. If $m = \log q / \log(1-\delta) = \Theta(-(\log q)/\delta)$, then with probability at most q measuring the $\mathcal{X}$ register of $|\varphi_1\rangle, \ldots, |\varphi_m\rangle$ will yield m times outcome 0.*

Proof.

$$\Pr\left[m \text{ times } 0 \,\middle|\, \forall j : \|P_0|\varphi_j\rangle\|^2 < 1-\delta\right] < (1-\delta)^m = (1-\delta)^{\log q/\log(1-\delta)} = q \ .$$

□

Now all the ingredients for wrapping up the argument are at hand; first consider $f \in L$. Let $S := \{s : f(x) = f(x \oplus s) \ \forall x\}$ be the set of all "Simon promises" of f, and $S^\perp := \{z : z \cdot s = 0 \ \forall s \in S\}$ the vectors that are orthogonal to all such promises. By Lemma 13 the nonzero z computed by the algorithm lie in $S^\perp$ and are linearly independent, therefore after $\dim S^\perp$ rounds of **for** loop in SimonTester, we measure $z = 0^n$ with certainty. Since $f \in L$, $\dim S > 0$ and thus $\dim S^\perp < n$.

If f is εn-far from being in L, then by Lemma 15 f is εn-far from being close to a function for which a $k < n$ and $z_1, \ldots, z_k$ exist so that f is constant when restricted to D_c for any of the $c \in \{0,1\}^k$. Therefore, by Lemma 14 case 2, for all $k < n$, $\|P_0|\psi\rangle\|^2 < 1 - \varepsilon^2/2$. Thus, Lemma 16 guarantees that we accept with probability at most $1/3$ if we let $q = 1/(3n)$ and thus $m = \mathrm{O}((\log n)/\varepsilon^2)$.

This concludes the proof of Theorem 7. □

5 Quantum Lower Bounds

In this section we prove that not every language has a fast quantum property tester.

Theorem 17 *Most properties containing $2^{n/20}$ elements of $\{0,1\}^n$ require quantum property testers using $\Omega(n)$ queries.*

Proof. Fix n, a small ε, and a quantum algorithm A making $q := n/400$ queries. Pick a property P as a random subset of $\{0,1\}^n$ of size $2^{n/20}$. Let

$$P_\varepsilon := \{y : d(x,y) < \varepsilon n \text{ for some } x \in P\}$$

where $d(x,y)$ denotes the Hamming distance between x and y. Using $\sum_{k=0}^{\varepsilon n} \binom{n}{k} \leq 2^{H(\varepsilon)n}$ for

$$H(\varepsilon) := -\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon) \ ,$$

we obtain $|P_\varepsilon| \leq 2^{(1/20+H(\varepsilon))n}$. In order for A to test properties of size $2^{n/20}$, it needs to reject with high probability on at least $2^n - 2^{(1/20+H(\varepsilon))n}$ inputs; but then, the probability that A accepts with high probability on a random $x \in \{0,1\}^n$ is bounded by $2^{(1/20+H(\varepsilon))n}/2^n$ and therefore the probability that A accepts with high probability on $|P|$ random inputs is bounded by

$$2^{-(1-1/20-H(\varepsilon))n|P|} = 2^{-2^{n/20+\Theta(\log n)}} \ .$$

We would like to sum this success probability over all algorithms using the union bound to argue that for most properties no algorithm can succeed. However, there is an uncountable number of possible quantum algorithms with arbitrary quantum transitions. But by Beals et al. [11], the acceptance probability of A can be written as a multilinear polynomial of degree at most $2q$ where the n variables are the bits of the input; using results of Bennett, Bernstein, Brassard, and Vazirani [12] and Solovay and Yao [39], every quantum algorithm can be approximated by another algorithm such that the coefficients of the polynomials describing the accepting probability are integers of absolute value less than $2^{n^{\mathrm{O}(1)}}$ over some fixed denominator. There are less than $2^{nH(2q/n)}$ degree-$2q$ monomials in n variables, thus we can limit ourselves to $2^{n^{\mathrm{O}(1)}2^{nH(2q/n)}} \leq 2^{2^{(n/20)\cdot(91/100)+\Theta(\log n)}}$ algorithms.

Thus, by the union bound, for most properties of size $2^{n/20}$, no quantum algorithm with q queries will be a tester for it. □

We also give an explicit natural property that requires a large number of quantum queries to test. We will make use of the following lemma:

Lemma 18 (see [2]) *Suppose $n = 2^k - 1$ and $d = 2t + 1 \leq n$. Then there exists a multiset of n bit strings $P = P(n,d) \subseteq \{0,1\}^n$ of size $2(n+1)^t$ such that under the uniform distribution over P, the Boolean random variables $\xi_1, \ldots, \xi_n$ that are the projection of $\xi \in P$ on its coordinates, are d-wise independent, each taking the values 0 and 1 with probability $1/2$.*

The proof of Lemma 18 is constructive and the construction uniform in n. For given n and d, consider the language $P(n,d)$ of n-bit strings, where $P(n,d)$ is the range of n Boolean d-wise independent variables, as asserted by the lemma. Classically, deciding membership in $P(n,d)$ takes more than d queries: for all d positions $i_1, \ldots, i_d$ and every string $v \in \{0,1\}^d$ there is a $z \in P(n,d)$ whose restriction to $i_1, \ldots, i_d$ is v. On the other hand, $\lfloor \log |P| \rfloor + 1 = \mathrm{O}(d \log n)$ queries are always sufficient.

Theorem 19 *Let $d \leq n/\log n - 1$ be odd and let $P = P(n,d)$ be the range of a d-wise independent Boolean variable as asserted by Lemma 18. Then for constant $\varepsilon < 1/2$, any ε-quantum tester for P requires at least $(d+1)/2$ quantum queries.*

Proof. For a property $P \subseteq \{0,1\}^n$ let again $P_\varepsilon := \{y : d(x,y) < \varepsilon n \text{ for some } x \in P\}$. By [11], a quantum computer that ε-tests a property P with T queries gives rise to a degree-$2T$ multilinear n-variable polynomial $p(x) = p(x_1, \ldots, x_n)$ that approximates P in the sense that $|p(x) - f(x)| \leq 1/3$ for every $x \in P \cup (\{0,1\}^n \setminus P_\varepsilon)$. Let $p(x_1, ..., x_n)$ be the corresponding polynomial to a quantum ε-test for P. We show that there must be high-degree monomials in p by comparing the expectation of $p(x)$ for randomly chosen $x \in \{0,1\}^n$ with the expectation of $p(x)$ for randomly chosen $x \in P$.

By the definition of P_ε and Lemma 18 we have

$$|P_\varepsilon| \leq 2^{H(\varepsilon)n}|P| = \mathrm{O}(2^{H(\varepsilon)n + d\log n}) \ ,$$

Hence for $d = n/\log n - \omega(1/\log n)$ and $\varepsilon < 1/2$ we get that $|P_\varepsilon| = \mathrm{o}(2^n)$. Thus for x uniformly distributed over $\{0,1\}^n$ we have:

$$\mathrm{E}[p(x)] = \frac{|P_\varepsilon|}{2^n}\,\mathrm{E}[p(x)|x \in P_\varepsilon] + \left(1 - \frac{|P_\varepsilon|}{2^n}\right)\mathrm{E}[p(x)|x \notin P_\varepsilon] \leq 1/3 + \mathrm{o}(1)$$

On the other hand, by the properties of p above, for x distributed uniformly over P holds $\mathrm{E}[p(x)|x \in P] \geq 2/3$. Considering $p(x) = \sum_i \alpha_i m_i(x)$ as a linear combination of n-variable multilinear monomials m_i, we have, by the linearity of expectation, $\mathrm{E}[p(x_1, \ldots, x_n)] = \sum_i \alpha_i \mathrm{E}[m_i(x_1, \ldots, x_n)]$. But for every m_i of degree at most d, by the d-wise independence of the bits of each $x \in P$ it follows that $\mathrm{E}[m_i(x)|x \in P] = \mathrm{E}[m_i(x)|x \in U]$, where U is the uniform distribution on $\{0,1\}^n$. Thus p must contain monomials of degree greater than d in order for those two expectations to differ by $1/3 - \mathrm{o}(1)$. We conclude that the number of queries T is greater than $d/2$. □

6 Further Research

Our paper opens the door to the world of quantum property testing. Several interesting problems remain, including

- Can one get the greatest possible separation of quantum and classical property testing, i.e., is there a language that requires $\Omega(n)$ classical queries but only $\mathrm{O}(1)$ quantum queries to test?
- Are there other natural problems that do not have quantum property testers? We conjecture for instance that the language $\{uuvv : u, v \in \Sigma^*\}$ does not have a quantum property tester.
- Beals et al. [11] observed that any k-query quantum algorithm gives rise to a degree-$2k$ polynomial in the input bits that gives the acceptance probability of the algorithm; thus, a quantum property tester for P gives rise to a polynomial that is on all binary inputs between 0 and 1, that is at least 2/3 on inputs with the property P and at most 1/3 on inputs far from having the property P. Szegedy [40] suggested to algebraically characterize the complexity of classical testing by the minimum degree of such polynomials; as mentioned in the introduction, our results imply that this cannot be the case for classical testers. However, it is an open question whether quantum property testing can be algebraically characterized in this way.
- Related to the second and the third item is the following question about polynomials. Is there a property $P \subseteq \{0,1\}^n$ for which every quantum ε-tester requires at least $\Omega(n)$ queries but for which there is a polynomial of constant degree $p(x_1, ..., x_n)$ such that: $0 \leq p(x) \leq 1$ for every x and $p(x) \leq 1/3$ for x's that are ε-far form P while $p(x) \geq 2/3$ for every $x \in P$. Such a P will show that polynomial characterization of quantum property testing, as suggested above, is impossible. It will also require other means of proving quantum non-testability results.

We hope that further research will lead to a greater understanding of what can and cannot be tested with quantum property testers.

Acknowledgments

We thank Ronitt Rubinfeld for discussions and pointers on property testing.

References

[1] N. Alon. Testing subgraphs in large graphs. In *Proceedings of 42nd IEEE FOCS*, pages 434–441. IEEE, 2001.

[2] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7:567–583, 1986.

[3] N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. *SIAM Journal on Discrete Mathematics*, 16:393–417, 2003.

[4] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. In *Proceedings of 40th IEEE FOCS*, pages 656–666. IEEE, 1999.

[5] N. Alon, I. Newman, M. Krivelevich, and M. Szegedy. Regular languages are testable with a constant number of queries. In *Proceedings of 40th IEEE FOCS*, pages 645–655, 1999.

[6] N. Alon and A. Shapira. Testing subgraphs in directed graphs. *Journal of Computer and Systems Sciences*, 69:354–382, 2004.

[7] N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. In *Proceedings of 46nd IEEE FOCS*, pages 429–438, 2005.

[8] N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. In *Proceedings of 46nd IEEE FOCS*, pages 429–438, 2005.

[9] N. Alon and A. Shapira. Every monotone graph property is testable. In *Proceedings of 37th ACM STOC*, pages 128–137, 2005.

[10] N. Alon and J. H. Spencer. *The probabilistic method*, appendix A. Wiley-Interscience, second edition, 2000.

[11] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS 98.

[12] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[13] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[14] M. Blum, M. Luby, and R. Rubinfeld. Self-testing and self-correcting programs, with applications to numerical programs. *Journal of Computer and System Sciences*, 47:549–595, 1993.

[15] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. In *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems (ISTCS'97)*, pages 12–23, 1997.

[16] H. Buhrman, L. Fortnow, I. Newman, and H. Röhrig. Quantum property testing. In *Proceedings of 14th SODA*, pages 480–488, 2003.

[17] A. Czumaj and C. Sohler. Abstract combinatorial programs and efficient property testers. *SIAM Journal on Computing*, 34:580–615, 2005.

[18] W. van Dam, F. Magniez, M. Mosca, and M. Santha. Self-testing of universal and fault-tolerant sets of quantum gates. In *Proceedings of 32nd ACM STOC*, pages 688–696, 2000.

[19] F. Ergün, S. Kannan, S. Kumar, R. Rubinfeld, and M. Vishwanathan. Spot-checkers. *Journal of Computer and System Sciences*, 60(3):717–751, 2000.

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

[21] E. Fischer. Testing graphs for colorability properties,. In *Proceedings of 12th SODA*, pages 873–882, 2001.

[22] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *Journal of Computer and Systems Sciences*, 68:753–787, 2004.

[23] E. Fischer and I. Newman. Testing of matrix properties. In *Proceedings of 33rd ACM STOC*, pages 286–295, 2001.

[24] K. Friedl, F. Magniez, M. Santha, and P. Sen. Quantum testers for hidden group properties. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science*, 2003.

[25] O. Goldreich. Combinatorial property testing (a survey), 1998. Manuscript.

[26] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

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

[28] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. In *Proceedings of 42nd IEEE FOCS*, pages 460–469. IEEE, Nevada, 2001.

[29] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996.

[30] L. Hales and S. Hallgren. An improved quantum fourier transform algorithm and applications. In *Proceedings of 41st IEEE FOCS*, pages 515–525, 2000.

[31] P. Høyer. Fourier sampling. Private communication, 2001.

[32] D. Mayers and A. Yao. Quantum cryptography with imperfect apparatus. In *Proceedings of 39th IEEE FOCS*, pages 503–509, 1998.

[33] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[34] M. Parnas and D. Ron. Testing the diameter of graphs. *Random structures and algorithms*, 20:165–183, 2002.

[35] D. Ron. Property testing. In S. Rajasekaran, P. M. Pardalos, J. H. Reif, and J. D. P. Rolim, editors, *Handbook of Randomized Computing*, volume 9 of *Combinatorial Optimization*. Kluwer, 2001.

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

[37] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94.

[38] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

[39] R. Solovay and A. Yao, 1996. Manuscript.

[40] M. Szegedy. Private communication. 1999.

[41] A. C-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of 18th IEEE FOCS*, pages 222–227, 1977.