Functions that have read-twice constant width branching programs are not necessarily testable

Eldar Fischer[*] Ilan Newman[†] Jiří Sgall[‡]

October 21, 2003

Abstract

We construct a property on 0/1-strings that has a representation by a collection of width-three read-twice oblivious branching programs, but for which any two-sided ϵ-testing algorithm must query at least $\Omega(n^{\delta})$ many queries for some fixed ϵ and δ. This shows that Newman's result [18] cannot be generalized to read-k-times functions for $k > 1$. In addition, we exhibit a property that has also a representation by a CNF formula of constant clause size. Hence, the non-testability results extend to properties that in addition have small (constant size) 0-witnesses.

*Department of Computer Science, The Technion, Haifa 32000, Israel; part of this work was done while at NEC Research Institute, Princeton NJ, USA. Email: eldar@cs.technion.ac.il. Partially supported by a joint Haifa University – Technion grant, and by a Coleman – Cohen Academic Lectureship fund.

†Department of Computer Science, University of Haifa, Haifa 31905, Israel; part of this work was done while at NEC Research Institute, Princeton NJ, USA. Email: ilan@cs.haifa.ac.il. Partially supported by a joint Haifa University – Technion grant.

‡Mathematical Institute, AS CR, Žitná 25, CZ-11567 Praha 1, Czech Republic. Email: sgall@math.cas.cz. Partially supported by Institute for Theoretical Computer Science, Prague (project LN00A056 of MŠMT ČR) and grant A1019901 of GA AV ČR.

1 Introduction

Combinatorial property testing deals with the following relaxation of decision problems: Given a fixed property and an input x, one wants to decide whether x has the property or is 'far' from having the property. The general notion of property testing was first formulated by Rubinfeld and Sudan [20] and first studied for combinatorial objects by Goldreich, Goldwasser and Ron [15].

A property in this respect is a collection of 0/1 strings, and being far is measured by the *Hamming distance*, namely, in how many places should an input string x be changed so as to have the property. An ϵ-test is a randomized algorithm which with probability at least 2/3 distinguishes between the case that x has the property and the case that x has distance at least ϵn from any string having the property. More precisely, if the input x has the property, the algorithm is required to answer 'yes' with probability at least 2/3; if the input x of length n does not have the property and moreover no x' which differs from x at most at ϵn bits has the property, the algorithm is required to answer 'no' with probability at least 2/3. Note that this definition allows two-sided error; one-sided error algorithms are more restricted, being required in addition to answer 'yes' with probability 1 if x has the property.

A property is said to be $(\epsilon, q(\epsilon, n))$-testable if there is an ϵ-test that for every input x of size n queries at most $q(\epsilon, n)$ chosen bits of the input string. When a property $\mathcal{P}$ is $(\epsilon, q(\epsilon, n))$-testable with $q = q(\epsilon)$ (i.e., q is a function of ϵ only, and is independent of n), we then say that $\mathcal{P}$ is ϵ-testable; we say that $\mathcal{P}$ is testable if it is ϵ-testable for every $\epsilon > 0$.

Property testing has recently become quite an active research area, see [15, 16, 8, 5, 2, 3, 18, 10, 9] for an incomplete list. Apart from its theoretical appeal, and the many questions it involves, it is related to PAC learning, program checking [14, 7, 20], probabilistically checkable proofs [4], and approximation algorithms [15]. For surveys on property testing see [11, 19].

One of the main tasks that emerged in the field, following [15] and then [2, 3], is to identify natural collections of properties that are efficiently testable (in terms of the number of queries). Goldreich et al. [15] studied some classes of properties (mainly graph properties) and identified many properties that are testable. Newman [18], following Alon et al. [3], proved that any property that can be computed by an oblivious read-once constant-width branching program is testable, even with one-sided error.

Here we prove that this cannot be generalized to read-twice branching programs. We exhibit two properties that can be computed by a width 3 read-twice oblivious branching program, but for which any (possibly two-sided) ϵ-test requires at least n^δ queries, for some $0 < \epsilon < 1$ and $0 < \delta < 1$. The first property that is described is constructive, it is somewhat simpler than the second one and admits a simpler proof of non-testability. The existence of the second property involves a probabilistic argument – hence, in this sense, it is not constructive. On the other hand it serves as a counter example for the following 'folklore' question in the area of property testing.

A Boolean function f is said to have $O(1)$ size 0-witnesses if it can be represented by a CNF formula in which every clause is of size $O(1)$. Namely, f has a depth two circuit in which the top gate is $\wedge$ and the bottom gates are $\vee$, each having a constant number of variables and negations of variables. For such a function, for every input assignment of the variables for which the function evaluates to 0, there are some $O(1)$ variables for which their corresponding values is a 'proof' that the function is 0 (the variables in the corresponding violated clause). So far, all properties that were known to be hard for (two-sided error) testing were functions whose 0-witnesses were large.

Our second property shows a strong non-testability result for a function with $O(1)$ 0-witnesses, with the additional feature that it has a CNF representation with $O(1)$ clause-size and such that every variable appears in $O(1)$ clauses. An example for a function with constant size 0-witnesses that is hard to test is given in [12]; however, the current lower bound on the number of queries is only $\Omega(\log n)$. Very recently, [6] produced an example of a 3CNF function (but one that is not represented by a read-twice BP) for which the bound on ϵ-testing is $\Omega(n)$; it is interesting to note that both the property in [6] and the property presented in the following are essentially based on linear equations modulo 2.

A preliminary version of part of the results presented here appeared in [13].

2 Preliminaries and notation

We identify properties with the collection of their characteristic Boolean functions. Namely, a property $\mathcal{P} \subseteq \{0,1\}^*$ is identified with $\{f_n : \{0,1\}^n \longrightarrow \{0,1\}\}$ such that $f_n(x) = 1$ if and only if $x \in \mathcal{P} \cap \{0,1\}^n$. For $x, y \in \{0,1\}^n$ we define $dist(x,y) = Hamming(x,y) = |\{i \mid x_i \neq y_i\}|$. For a property $\mathcal{P} \subseteq \{0,1\}^n$, and $x \in \{0,1\}^n$ we define $dist(x,\mathcal{P}) = \min_{y \in \mathcal{P} \cap \{0,1\}^n} dist(x,y)$. For $0 < \epsilon < 1$, we say that x is ϵ-far from $\mathcal{P}$ if $dist(x,\mathcal{P}) \geq \epsilon n$.

An oblivious leveled branching program (BP) here is a directed graph B, in which the nodes are partitioned into levels $L_1, \ldots, L_{m+1}$ and which satisfies the following. There are two special nodes; a 'start' node belonging to L_1 and an 'accept' node belonging to L_{m+1}. All edges are between nodes of consecutive levels, and each node apart from those on L_{m+1} has exactly two outgoing edges to the next level, one of which is labeled by 0 and the other by 1. In addition, each level L_i apart from L_{m+1} is associated with a member l_i of $\{1, \ldots, n\}$, that refers to a location of a bit in the input. An input $x \in \{0,1\}^n$ naturally defines a path in B: It contains one node from each level, starting with the start-node at L_1; given the choice of a node from a level L_i, $1 \leq i \leq m$, its outgoing edge which is labeled by x_{l_i} is followed to select the node at L_{i+1}. A leveled BP defines naturally a Boolean function $g : \{0,1\}^n \longrightarrow \{0,1\}$ in the following way: $g(x) = 1$ if the path that x defines reaches *accept*, otherwise $g(x) = 0$. The *width* of a leveled BP is the cardinality of its largest level.

A *read-k-times* BP for a function $g : \{0,1\}^n \longrightarrow \{0,1\}$ is an oblivious leveled branching program computing g, in which no member of $\{1, \ldots, n\}$ is associated with more than k of the levels $L_1, \ldots, L_m$.

In the following we denote by F_2 the field of two elements, and denote by F_m the field of m elements whenever such a field exists. We denote by $\oplus$ the addition in F_2. For two subsets A and B, with some abuse of notation, we also use $A \oplus B$ to denote their symmetric difference, $A \oplus B = (A \setminus B) \cup (B \setminus A)$. We omit all floor and ceiling signs whenever the implicit assumption that a quantity is integer makes no essential difference. We make no attempt to optimize the coefficients involved, just the function types (e.g., polynomial versus logarithmic).

3 The main result

Theorem 3.1 *There exist fixed $\epsilon, \delta > 0$ and a property $\mathcal{P}$ that has a width 3, read-twice oblivious branching program, for which any ϵ-test must query n^δ queries even if we allow two-sided error.*

We present two properties that serve as a proof for Theorem 3.1. The first, which is the simpler of the two and whose existence proof is constructive, is discussed in Section 4. The second one, which has the extra property of having 0-witnesses of constant size, is discussed in Section 5.

Both properties are based on some fixed Boolean circuits over F_2 with $\oplus$ gates. We now describe the basic definitions and preliminaries mutual to both constructions.

Let $G = (V, E)$ be a directed acyclic graph with $V_1 \subseteq V$ being the set of nodes of in-degree 0. For each $v \in V \setminus V_1$ let $N_{in}(v) = \{u \in V \mid (u, v) \in E\}$, namely $u \in N_{in}(v)$ if and only if there is a directed edge from u to v. In cases where we allow multiple edges, the above will be a multi-set, and we will use the usual extensions of the following definitions to multi-sets as well.

We associate a Boolean circuit C_G with a graph G as follows. A Boolean variable X_v is associated with each $v \in V_1$. For every Boolean assignment to the formal variables, a Boolean value is associated with each $v \in V \setminus V_1$, that is equal to the parity of the values that are associated with the vertices of $N_{in}(v)$. In particular each vertex in $V \setminus V_1$ is associated with a linear function (over F_2) in the formal Boolean variables $\{X_v \mid v \in V_1\}$. Clearly, for every Boolean assignment to the variables associated with V_1, the circuit C_G defines Boolean values for every other vertex in the circuit. Such circuits are called *$\oplus$-circuits*.

In the following, $\{0, 1\}^V$ denotes the set of all Boolean functions whose domain is V. The circuit C_G is associated with a property $\mathcal{P}_G \subseteq \{0, 1\}^V$ in the following way: Suppose a Boolean value x_v is assigned to each node v. This assignment is said to have $\mathcal{P}_G$ if for every $v \in V \setminus V_1$, $x_v = \bigoplus_{u \in N_{in}(v)} x_u$. Namely, the property is satisfied if the assignments to the nodes represent a consistent computation of C_G, where $\{x_v \mid v \in V_1\}$ represent the values of the original formal variables $\{X_v \mid v \in V_1\}$.

$\mathcal{P}_G$ in general does not have a read-twice BP. However, it is associated with a semantically close property $\mathcal{P}'_G$ that has a read-twice BP. We add a new Boolean value $x_{(u,v)}$ for every edge $(u, v) \in E$; we think of the edge values as carrying the Boolean values between the nodes of the circuit. The property $\mathcal{P}'_G \subseteq \{0, 1\}^{V \cup E}$ is defined as follows: An assignment has property $\mathcal{P}'_G$ if (i) for every edge $(v, w) \in E$ we have $x_v = x_{(v,w)}$, and (ii) for every node $v \in V \setminus V_1$ we have $x_v = \bigoplus_{(u,v) \in E} x_{(u,v)}$.

Claim 3.2 *Assume that for all $v \in V \setminus V_1$, $|N_{in}(v)| \leq \Delta$. Then $\mathcal{P}'_G$ has the following properties.*

(1) If there is an (ϵ, q)-test for $\mathcal{P}'_G$ then there is also a $((\Delta + 1)\epsilon, q)$-test for $\mathcal{P}_G$.

(2) $\mathcal{P}'_G$ has a width 3 oblivious read-twice BP.

(3) $\mathcal{P}'_G$ can be expressed by a CNF formula in which every clause is of size at most $\Delta + 1$, and every variable appears in at most $4 + 2^\Delta$ clauses.

Proof: (1) We define a mapping $\rho : \{0, 1\}^V \longrightarrow \{0, 1\}^{V \cup E}$ that maps inputs for $\mathcal{P}_G$ to inputs for

$\mathcal{P}'_G$: For every $x \in \{0,1\}^V$, $\rho(x) \in \{0,1\}^{V \cup E}$ is the assignment that assigns v the value x_v for every $v \in V$ and assigns (u,v) the value x_u for every $(u,v) \in E$. If T is an (ϵ, q)-test for $\mathcal{P}'_G$ then it can be used to define a test T' for $\mathcal{P}_G$ as follows: for every $x \in \{0,1\}^V$ we apply the test T on $\rho(x)$ (i.e., every time (u,v) is queried we reply with x_u and every time v is queried we reply with x_v). The number of queries in T' is the same as in T. If $x \in \mathcal{P}_G$ then obviously $\rho(x) \in \mathcal{P}'_G$. If, on the other hand, $dist(x, \mathcal{P}_G) \geq (\Delta+1)\epsilon \cdot |V|$, then $dist(\rho(x), \mathcal{P}'_G) \geq (\Delta+1)\epsilon \cdot |V| \geq \epsilon \cdot |V \cup E|$, and hence T' is ϵ-far from $\mathcal{P}'_G$.

(2) We simply check, for each v, (i) and (ii) from the definition of $\mathcal{P}'_G$. Each of these checks can easily be done by a read-once width-two branching program; overall we need a program of width three, where the third branch just collects all the negative results of the checks. Every node value x_v is read only by its comparison program and its parity verification program (once by each program), and every edge value $x_{(v,w)}$ is read only by the comparison program of v and the parity verification program of w.

(3) The resulting CNF is a conjunction of CNF's checking (i) and (ii) from the definition of $\mathcal{P}'_G$. We must be a bit careful in checking (i) to achieve a bounded number of occurrences of each variable. For a vertex v with outgoing edges (v, w_i), $i = 1, \ldots, t$, ordered arbitrarily, we check $x_v = x_{(v,w_1)}$ and $x_{(v,w_i)} = x_{(v,w_{i+1})}$ for $i = 1, \ldots, t-1$. This results in at most 4 clauses of size 2 for each variable. Each parity check (ii) is equivalent to a conjunction of 2^Δ clauses of size $\Delta + 1$. ■

We need the following well-known tool.

Lemma 3.3 *Let $x_1, \ldots, x_k$ be random Boolean variables and suppose that for every $\emptyset \neq S \subset [1, \ldots, k]$, the probability of $\bigoplus_{i \in S} x_i = 1$ is exactly $\frac{1}{2}$. Then the distribution on the values of $x_1, \ldots, x_k$ is the uniform distribution on all k-bit Boolean strings.* ■

Let $G(U, V; E)$ be a bipartite graph, and let $V' \subseteq U \cup V$. We denote by $N(V')$ the set of all neighbors of all the vertices in V' and denote by $O_G(V')$ the set of vertices $u \in N(V')$ for which $|\{(u,v) \mid v \in V'\}|$ is odd. Namely, $O_G(V')$ are those vertices in that have an odd number of neighbors in V'.

The motivation behind this definition is the following. We associate each subset of vertices $S \subseteq V$ (similarly for $T \subseteq U$) with the corresponding characteristic vector $x_S \in \{0,1\}^V$. Then G defines a linear map $A_G : \{0,1\}^U \longrightarrow \{0,1\}^V$ over F_2, by $A_G(x) = (y_v \mid v \in V)$ where $y_v = \bigoplus_{u \in V | (u,v) \in E} x_u$. In this map, viewed as a map between sets, the set $S \subseteq U$ is mapped to the set $O_G(S)$. If we direct all edges from U to V and look at the $\oplus$-circuit that is defined by G, then the map between input assignments to output values (viewed as mapping between column vectors over F_2) is exactly the map A_G above.

Definition 3.4 *Let $G(U, V; E)$ be a bipartite graph where $|V| = |U| = n$, and let $\mathcal{F}$ be a family of nonempty sets, each a subset of either V or U. G is said to be $(\alpha, \mathcal{F})$-odd-expanding if for every $V' \in \mathcal{F}$, $|O_G(V')| \geq \min((1+\alpha)|V'|, \alpha n)$.*

Unlike the case with the usual notion of expanders, it is easy to show that there exists no "odd expander" that works for all nonempty subsets of the vertices. To see this, consider the map A_G

defined above. If this map is 1-1 then certainly there exists a set whose image is a singleton (and thus its size cannot increase), while if it is not 1-1, then there is a non-empty set that is mapped to the empty set (namely any set that corresponds to a vector in the kernel of this map), which means that the image of some sets may actually vanish.

4 The first property

In this section we construct a simple bipartite graph $G = (U, V; E)$ for which $\mathcal{P}_G$ is not testable. The construction is simple, but the cost we pay for that is that the degrees are no longer constant. Thus we cannot directly rely on Claim 3.2 to consider $\mathcal{P}'_G$ as our non-testable property, as the approximation parameter ϵ would decrease too much. Instead we use the same method but with some extra padding at the output vertices to construct a property $\mathcal{P}''_G$ that is also non-testable, and that can be represented by a constant depth read-twice BP.

Let $G = (U, V; E)$ be a regular balanced bipartite graph with $|U| = |V| = n$ such that $deg(v) = \Delta$ for every $v \in U \cup V$. In the following we will use $\Delta = \Theta(\sqrt{n})$. We direct all edges from U to V. Clearly the resulting graph will be acyclic. Now $\mathcal{P}_G$ is defined on strings of length $n' = 2n$.

We define a property $\mathcal{P}''_G$ over Boolean assignments to $n'' = 2n(\Delta + 1)$ variables $\{x_u \mid u \in U\} \cup \{y_v \mid v \in V\} \cup \{x_{(u,v)}, y_{(u,v)} \mid (u, v) \in E\}$. An assignment will have the property $\mathcal{P}''_G$ if (i) for every $u \in U$, all $\Delta + 1$ variables x_u and $x_{(u,v)}$, $v \in N(u)$ have the same value, (ii) for every $v \in V$, we have $y_v = \bigoplus_{(u,v)\in E} x_{(u,v)}$, and (iii) for every $v \in V$, all $\Delta + 1$ variables y_v and $y_{(u,v)}$, $u \in N(v)$, have the same value.

Claim 4.1 *$\mathcal{P}''_G$ has the following properties.*

(1) If there is an (ϵ, q)-test for $\mathcal{P}''_G$ then there is also an (ϵ, q)-test for $\mathcal{P}_G$.

(2) $\mathcal{P}''_G$ has a width 3 oblivious read-twice BP.

Proof: (1) Assume that there is an (ϵ, q)-test for $\mathcal{P}''_G$. We define a mapping from inputs to the test for $\mathcal{P}_G$ to inputs for $\mathcal{P}''_G$, similarly to Claim 3.2. For every $w = (x', y') \in \{0, 1\}^{U \cup V}$ let $\rho(w)$ be defined as follows: The value of every variable x_u and $x_{(u,v)}$ is x'_u, and the value of every variable y_v and $y_{(u,v)}$ is y'_v. It is obvious that for every $w \in \mathcal{P}_G$, $\rho(w) \in \mathcal{P}''_G$. In addition, if $dist(x, \mathcal{P}_G) \geq \epsilon \cdot 2n$ then $dist(x, \mathcal{P}''_G) \geq \epsilon \cdot 2n(\Delta + 1)$. This is true as all the $\Delta + 1$ values of the copies of each bit of w are equal for an input to belong to $\mathcal{P}''_G$. To ϵ-test $\mathcal{P}_\mathcal{G}$ on an input $w = (x', y')$, we perform T on $\rho(w)$. Every time that x_u or $x_{(u,v)}$ is queried in $\rho(w)$ we just query x'_u and every time that y_v or $y_{(u,v)}$ is queried in $\rho(w)$ we query y'_v. This results in a valid test for $\mathcal{P}_G$ with at most q queries.

(2) Just like the proof for $\mathcal{P}'_G$ in Claim 3.2, we check all the conditions (i)-(iii) from the definition. Every variable appears in at most two such checks, so the resulting BP is read-twice. ∎

Our aim now is to construct a family of graphs G_n for an infinite sequence of n for which $\mathcal{P}_G$ is hard to test.

Let $P = (U, V)$ be a finite projective geometry where U is a set of $n = m^2 + m + 1$ points and V is a set of n lines, with m being an odd prime. For the construction of these well-known structures one can consult for example the chapter about designs in [21], or read [17] for an in-depth (algebraic) treatment. We shall use only the following basic regularity properties of these finite geometries: every line contains exactly $m+1$ points, every point is contained in exactly $m+1$ lines, and for every two points there is exactly one line containing both of them. Let $G = (U, V; E)$ be the bipartite graph that corresponds to the incidence relation of P, namely in which $(u, v) \in E$ if the point u is contained in the line v. This defines $\mathcal{P}_G$ as above. We will shortly prove the following.

Theorem 4.2 *Any (2 sided error) $1/20$-test for $\mathcal{P}_G$ requires $m + 2 = \Omega(n^{1/2})$ queries.*

Its corollary which in particular proves Theorem 3.1 is the following.

Theorem 4.3 *Any (2 sided error) $1/20$-test for $\mathcal{P}''_G$ requires $\Omega(n^{1/3})$ queries, where n is the input length.*

Proof: Immediate from Theorem 4.2, Claim 4.1, and the fact that if the input length of $\mathcal{P}_G$ is n then the input length for the corresponding $\mathcal{P}''_G$ is $\Theta(n^{3/2})$. ∎

Proof of Theorem 4.2: We use Yao's method, [22]. We shall define a distribution A (for 'accepted') on inputs that satisfy $\mathcal{P}_G$, a distribution N (for 'non-acceptable') on inputs that are far from satisfying it, and a distribution $D = 0.5A + 0.5N$. Namely, we choose with probability $1/2$ either A or N respectively, and then choose an input according to the chosen distribution. Let T be any deterministic decision tree that makes only $q \leq m+1$ queries and supposedly distinguishes between the case that that an input belongs to $\mathcal{P}_G$ and the case that an input is $1/20$-far from satisfying $\mathcal{P}_G$. We prove that T errs with probability at least $1/3$ over the distribution D on inputs, even if it is allowed to give any answer for inputs which do not fall into either of the two cases above.

The distribution A is defined as follows: We first choose a random assignment on U, $x_u \in_R \{0, 1\}$ for every $u \in U$. Then, for every $l \in L$, we let $y_l = \bigoplus_{u \in l} x_u$. In other words, y_l is the Boolean value that the circuit that corresponds to G_P is computing on l for the input assignment on U. By definition, A is concentrated on assignments in $\mathcal{P}_G$.

The distribution N is defined as follows: We first choose a random assignment on U, $x_u \in_R \{0, 1\}$, for every $u \in U$, as before. Then we choose a random assignment on L independent of the 'input' assignment on U. Namely, we choose $w = (x, y)$ according to the uniform distribution R on $2n$-bit binary sequences. We then let N be the distribution of the above, conditioned on the event that the input w thus generated is $1/20$-far from $\mathcal{P}_G$.

The following asserts that N is well approximated by R. Note that this lemma does not use the properties of projective planes.

Lemma 4.4 *Let $w \in \{0, 1\}^{2n}$ be an assignment that is generated by the uniform distribution R. Then, for any balanced bipartite graph G, $\mathrm{Prob}_{w \in R}[dist(w, \mathcal{P}_G) \leq n/10] = o(1)$.*

Proof: For any (x', y') with $dist((x', y'), \mathcal{P}_G) \leq \epsilon n$, there exists $(x, y) \in \mathcal{P}_G$ such that both $dist(x', x) \leq \epsilon n$ and $dist(y', y) \leq \epsilon n$. Now consider a fixed $w = (x, y)$ which has the property $\mathcal{P}_G$ and count the number of strings (x', y') with $dist(x', x) \leq \epsilon n$ and $dist(y', y) \leq \epsilon n$. The number of eligible values for x' as well as y' is bounded by $2^{H(\epsilon)\cdot n}$, where $H(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$ is the entropy function. Since the number of $(x, y) \in \mathcal{P}_G$ is 2^n, the probability that random (x', y') satisfies $dist((x', y'), \mathcal{P}_G) \leq \epsilon n$ is at most $2^n \cdot (2^{H(\epsilon)\cdot n})^2 / 2^{2n} = 2^{(2H(\epsilon)-1)n}$. For $\epsilon < 1/10$, $H(\epsilon) < 1/2$ and thus the probability is $o(1)$. ■

The following lemma show that small subsets of bits of strings generated from A are independent. This later implies that with a few queries, strings from A are indistinguishable from those from R.

Lemma 4.5 *The bits of strings $w = (x, y)$ from the distribution A are $(m+1)$-wise independent.*

Proof: Consider $U' \subseteq U$ and $V' \subseteq V$ such that $|U' \cup V'| \leq m + 1$. We prove by induction on $|V|$ that the bits x_u, $u \in U'$, and y_v, $v \in V'$, are independent. If $V = \emptyset$ then this follows trivially from the definition of the distribution A.

Otherwise, fix an arbitrary $v' \in V'$. We claim that there exists $u' \in N(v')$ such that $u' \notin U'$ and for all $v \in V \setminus \{v'\}$, $u' \notin N(v)$: By the properties of projective planes, for every $v \in V' \setminus \{v'\}$, $|N(v) \cap N(v')| \leq 1$. Thus the condition excludes at most $|U' \cup V \setminus \{v'\}| = m - 1$ candidates for u' out of the m elements of $N(v')$, and hence u' exists.

Now consider random values for all x_u, $u \neq u'$. These determine all the values of x_u, $u \in U'$, and y_v, $v \in V' \setminus \{v'\}$, and by the induction assumption these bits are independent. For any fixed values of x_u, $u \neq u'$, the bit $y_{v'}$ is still a random bit, as it depends on $x_{u'}$. Thus all bits x_u, $u \in U'$, and y_v, $v \in V'$, are independent. This completes the inductive step and the proof. ■

Let T be any deterministic decision tree that makes $q \leq m + 1$ queries in the worst case, let $T(w)$ be the output of T on input w.

For a given leaf l of T, the queried bits are independent random variables both in the distribution A, by Lemma 4.5, and for R by the definition of R. Thus $\mathrm{Prob}_{w\in A}[w \text{ reaches } l] = \mathrm{Prob}_{w\in R}[w \text{ reaches } l]$, which implies $\mathrm{Prob}_{w\in A}[T(w) = 1] = \mathrm{Prob}_{w\in R}[T(w) = 1]$ and thus also

$$\mathrm{Prob}_{w\in A}[T(w) = 0] + \mathrm{Prob}_{w\in A}[T(w) = 1] = \mathrm{Prob}_{w\in A}[T(w) = 0] + \mathrm{Prob}_{w\in R}[T(w) = 1] = 1 \quad (1)$$

Let F be the event that for a $w \in R$, $dist(w, \mathcal{P}_G) \geq n/10$. The probability of an error of T on a random input from D is at least

$$\begin{aligned}
&\frac{1}{2}\mathrm{Prob}_{w\in A}[T(w) = 0] + \frac{1}{2}\mathrm{Prob}_{w\in N}[T(w) = 1] \\
= \; &\frac{1}{2}\mathrm{Prob}_{w\in A}[T(w) = 0] + \frac{1}{2}\frac{\mathrm{Prob}_{w\in R}[T(w) = 1 \wedge F]}{\mathrm{Prob}_{w\in R}[F]} \\
\geq \; &\frac{1}{2}(\mathrm{Prob}_{w\in A}[T(w) = 0] + (1 + o(1))\mathrm{Prob}_{w\in R}[T(w) = 1]) - \frac{1}{2}(1 + o(1))(1 - \mathrm{Prob}_{w\in R}[F]) \\
= \; &\frac{1}{2} - o(1),
\end{aligned}$$

The inequalities are by Lemma 4.4 and equation (1). ■

5 The second property

In this section we construct a $\oplus$-circuit G (for infinitely many sizes n) for which the property $\mathcal{P}'_G$ is non-testable. We will have here the extra feature that the in-degree of the circuit will be constant. Hence, by Claim 3.2, this property has the extra feature of having $O(1)$ 0-witnesses (so it is a CNF with $O(1)$-clause size). Also, by the same claim, proving the lower bound on the complexity of $\mathcal{P}_G$ immediately implies the bound $\mathcal{P}'_G$.

To define the corresponding graph, we first define the following structure. Let $V = [n] \times [m]$. We think of V as the disjoint union of m disjoint vertex sets of size n, $V_1, \ldots, V_m$. Let $G_i = (V_i, V_{i+1}; E_i)$, $i = 1, \ldots, m-1$, be a sequence of bipartite graphs. This naturally defines a graph on V by letting the induced subgraph on $V_i \cup V_{i+1}$ be G_i. We then define the directed acyclic graph $\mathcal{G}$ on V by directing all edges from V_i to V_{i+1}, $i = 1, \ldots, m-1$.

The property $\mathcal{P}_{\mathcal{G}}$ depends on the exact choice of the collection $\mathcal{G} = \{G_1, \ldots, G_{m-1}\}$. The following theorem, whose proof is provided in Section 5.1, asserts that there exists $\mathcal{G} = \{G_1, \ldots, G_{m-1}\}$ for which $\mathcal{P}_{\mathcal{G}}$ is not testable.

Theorem 5.1 *For $m = \frac{1}{2}n^{1/4}$ there exists a family of graphs $\mathcal{G} = \{G_1, \ldots, G_{m-1}\}$ with $\Delta(\mathcal{G}) \leq 7$, for which any algorithm that $5 \cdot 10^{-7}$-tests the property $\mathcal{P}_{\mathcal{G}}$ defined above requires $\Omega(n^{1/8})$ queries.*

Note that the total input size (the total number of variables) in both $\mathcal{P}_{\mathcal{G}}$ and $\mathcal{P}'_{\mathcal{G}}$, for the set of graphs $\mathcal{G}$ that is guaranteed by Theorem 5.1, is $\Theta(nm) = \Theta(n^{5/4})$.

5.1 Proof of Theorem 5.1

For $\mathcal{P}_{\mathcal{G}}$ (and hence $\mathcal{P}'_{\mathcal{G}}$) not to be efficiently testable, we need a family $G_1, \ldots, G_{m-1}$ of bipartite graphs that are good odd-expanders. We will first develop the machinery and prove that it implies the non-testability result. In Section 5.2 we will prove that bipartite graphs with the required properties exist.

In the rest of the sequel let $V_1, \ldots, V_m$ be disjoint sets of size n. For every $1 \leq i < m$, we let G_i be a bipartite graph with the vertex set $V_i \cup V_{i+1}$. We now define the notion of (parity) propagations. Given two vertices $u \in V_i$ and $v \in V_j$ for some $1 \leq i \leq j \leq m$, a *propagation path* from u to v is directed path from u to v in $\mathcal{G}$, namely, it is a sequence of vertices $v_k \in V_k$, $i \leq k \leq j$, such that $v_i = u$, $v_j = v$, and for every $i \leq k < j$ the pair (v_k, v_{k+1}) is an edge of G_k. For $1 \leq i \leq m$ and a set S of vertices from $V_1 \cup \ldots \cup V_m$, we define the *propagation* of S into V_i as the set of vertices $P_i(S) \subseteq V_i$ in the following way: $v \in P_i(S)$ if and only if there is an *odd* number of distinct propagation paths from vertices of S that end in v (for vertices of S that do not belong to $V_1 \cup \ldots \cup V_i$, there are simply zero propagation paths from them to vertices in V_i). Similarly, for a set $T \subset V_1 \cup \ldots \cup V_m$ we define the *backpropagation* of T into V_i as the set of vertices $BP_i(S) \subseteq V_i$

that contains $v \in V_i$ if and only if there is an odd number of propagation paths from v that end in T. For every vertex y we define $BP(y) = BP_1(\{y\})$ namely, $BP(y)$ is the backpropagation of the singleton y into V_1. Let $ch(y)$ be the characteristic vector of $BP(y)$, considered as a member of $(F_2)^n$.

Some intuition about propagations can be given by the following: If $S \subseteq V_j$ for some j then its propagation into V_j is S itself, while its propagation into V_{j+1} is just $O_{G_j}(S)$. Its propagation into V_{i+1} for $i \geq j$ can be calculated recursively as $O_{G_i}(P_i(S))$. For $S \subseteq V_1 \cup \ldots \cup V_m$, the propagation of S into V_i is equal to $\bigoplus_{j=1}^{i} P_i(S \cap V_j)$.

More about the meaning of the propagations and backpropagations is given by the following observations. To state them, we consider a member of $\mathcal{P}_{\mathcal{G}}$, given by a value x_v for every $v \in V$.

Observation 5.2 *For every i let $U_i = \{v \in V_i \mid x_v = 1\}$. Then U_i is exactly the propagation of U_1 into V_i for every i.*

Proof: By a simple induction using the connection of propagations to the notion of $O_{G_i}(U_i)$. ■

Observation 5.3 *Let $v \in V$, and let $BP(v)$ be the backpropagation of $\{v\}$ into V_1. Then $x_v = \bigoplus_{u \in BP(v)} x_u$.*

Proof: By induction on the i such that $v \in V_i$ (using also the fact that the backpropagation of a set S, $S \subset V_j \cup \ldots \cup V_m$, into V_i equals the backpropagation of its backpropagation to V_j for any $i < j$). ■

The last observation is easy to expand to (note that the following also holds for an S that has vertices from more than one level):

Observation 5.4 *Let $S \subset V$ and let $BP(S)$ be the backpropagation of S into V_1. Then $\bigoplus_{v \in S} x_v = \bigoplus_{u \in BP(S)} x_u$.* ■

An analogue of Observation 5.4 to the characteristic vectors is also true:

Observation 5.5 *Let $S \subset V$ and let $\{ch(y) \mid y \in S\}$ be the set of the characteristic vectors associated with the vertices of S. Let $ch(BP(S))$ be the characteristic vector of the backpropagation of S into V_1. Then $\sum_{y \in S} ch(y) = ch(BP(S))$ (note that the sum in the left hand side is of vectors over F_2).*

Proof: The proof is by induction on the size of S. For $S = \{y\}$ this is just the definition of $ch(y)$.

For $S = S' \cup \{y\}$, $y \notin S'$, let $v \in V_1$ be in $BP(S') \cap BP(y)$. Then there is an odd number of paths from v to S' and an odd number of paths from v to y, and hence an even number of paths from v to $S' \cup \{y\}$. Hence $v \notin BP(S)$. Similar arguments for the other three cases concerning containment in $BP(S')$ and in $BP(y)$ show that indeed the coordinates that are 1 in $ch(BP(S))$ are exactly those that correspond to the vertices in $BP(S') \bigoplus BP(y)$. ■

The notion of propagation is central to what follows. In order to better understand it consider the circuit that is defined by $\mathcal{G}$ as performing successively the mappings A_{G_j}, $j = 1, \ldots, m-1$. Let $x_i = x(U_i)$ be the characteristic vector of the set U_i that was defined in Observation 5.2. In particular, x_1 is just the 'input vector' to the circuit. Then, observation 5.2 asserts that vector x_i is just the result of $A_{G_{i-1}} \cdot \ldots \cdot A_{G_1} \cdot x_1$. Observation 5.3 provides the specific linear equation for the 'coordinate' x_v inside x_i in terms of the vector x_1.

For the rest of the sequel let $(4.5^{2.5} \cdot e^{12.5})^{(2/3)} < 51000 \leq C$. Fix $r = 0.01\sqrt{m} = 0.01n^{1/8}$. Let $\mathcal{F} \subseteq 2^V$ for some set V of size n. We denote by $\mathcal{F}^{\leq r}$ the family of all subsets $S \subset V$ for which there exists $T \in \mathcal{F}$ such that $|(S-T) \cup (T-S)| \leq r$. Namely, $\mathcal{F}^{\leq r}$ contains all sets that can be obtained from sets of $\mathcal{F}$ by adding or deleting at most r elements.

Our aim now is to define the families $\mathcal{F}_i$ for which we want G_i to be odd-expanding.

Definition 5.6 *Suppose that $\mathcal{G} = G_1, \ldots, G_{m-1}$ is the family of bipartite graphs $G_i = (V_i, V_{i+1}, E_i)$.*

1. *Let $\mathcal{X}_m$ be the family of all nonempty subsets of V_m of size at most r. The sets $\mathcal{X}_i$, $1 \leq i \leq m-1$, are defined inductively ($\mathcal{X}_i$ from $\mathcal{X}_{i+1}$) by: Let $\mathcal{Y}_i = \{O_{G_i}(V) \mid V \in \mathcal{X}_{i+1}\}$ and let $\mathcal{X}_i = \mathcal{Y}_i^{\leq r} \setminus \{\emptyset\}$.*
2. *Let $\mathcal{Z}_1$ be the family of all nonempty subsets of V_1 of size at most $n/(10C)$. Each $\mathcal{Z}_i$, $i \geq 2$, is defined inductively to contain all nonempty subsets of V_i of size at most $n/(10C)$, and in addition all nonempty subsets that are the propagations of the members of $\mathcal{Z}_{i-1}$ into V_i.*
3. *Finally we set $\mathcal{F}_i = \mathcal{X}_{i+1} \cup \mathcal{Z}_i$ for every $1 \leq i \leq m-1$.*

Note that by our choice of parameters $|\mathcal{X}_m| \leq n^r$ and hence $|\mathcal{X}_i| \leq n^{r(m+1-i)} \leq n^{rm} \leq 2^{\sqrt{n}}$ for each i. Also, $|\mathcal{Z}_i| \leq \binom{n}{(n/(10C))} + |\mathcal{Z}_{i-1}|$ and hence $|\mathcal{Z}_i| \leq m\binom{n}{(n/(10C))} = o(e^{n/C})$ for every i, so $|\mathcal{F}_i| < e^{n/C}$ for every i, for sufficiently large n.

In the following discussion, we shall restrict ourselves to an instance of $\mathcal{G}$ where each G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding. The existence of such a setup is proven in Section 5.2.

Proposition 5.7 *Let $\mathcal{G} = \{G_1, \ldots, G_{m-1}\}$ where $G_i = (V_i, V_{i+1}, E_i)$, $i = 1, \ldots, m-1$, and assume that G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding. Then any $(\frac{1}{33C} - o(1))$-testing algorithm for $\mathcal{P}_{\mathcal{G}}$ requires at least $\Omega(n^{1/8})$ many queries (even for a two-sided error).*

For the proof of Proposition 5.7 we shall use Yao's method [22], as in the proof of Theorem 4.2. We now define the two distributions, A and N. We first select a uniformly random integer l such that $\frac{1}{3}m \leq l \leq \frac{2}{3}m$. We define A to be concentrated on $\mathcal{P}_{\mathcal{G}}$ – we choose independently uniformly and randomly the values of $\{x_u \mid u \in V_1\}$ and then extend them to an assignment to all variables as per the calculation depicted by $\mathcal{G}$. Although l was not used in choosing the input according to

A, we shall use it in the analysis that compares A to N, as the latter uses the random choice of l in choosing its input.

For defining N, we choose two inputs $\mathcal{A}$ and $\mathcal{A}'$ that satisfy the property. $\mathcal{A}$ is chosen according to A. To choose $\mathcal{A}'$ we take an arbitrary member z of $\{x_u \mid u \in V_1\}$, and invert its value with respect to its value in $\mathcal{A}$, keeping all other values of $\{x_u \mid u \in V_1\}$. The rest of $\mathcal{A}'$ is again the extension of these values. To choose an input $\mathcal{B}$ according to N, we first choose l, $\mathcal{A}$ and $\mathcal{A}'$ as above, and let $\mathcal{B}$ be identical to $\mathcal{A}$ for any value x_v, $v \in V_1 \cup \ldots \cup V_l$, and identical to $\mathcal{A}'$ for any value x_w, $w \in V_{l+1} \cup \ldots \cup V_m$. Finally let D be the distribution on all inputs defined by $D = 0.5A + 0.5N$. We next prove that an input chosen according to N is indeed far from satisfying $\mathcal{P}_{\mathcal{G}}$.

Claim 5.8 *Any input chosen according to N is $\frac{nm(1-o(1))}{33C}$-far from satisfying $\mathcal{P}_{\mathcal{G}}$.*

Proof: Let B be chosen as above by choosing l, z, $\mathcal{A}$ and $\mathcal{A}'$. We first note that $\mathcal{A}'$ differs from $\mathcal{A}$ in all vertices that are in the propagation of z into any V_i. Since we assume that each G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding this means that for every $i \geq \log_{1+C}(n/C)$, the propagation set of z is of size at least n/C. Hence for every $i \geq \Theta(\log n)$, $\mathcal{A}$ differs from $\mathcal{A}'$ in at least n/C places.

We then note that $\mathcal{B}$ is at least $\frac{mn(1-o(1))}{3C}$-far from both $\mathcal{A}$ and $\mathcal{A}'$. This is true as $\mathcal{B}$ is identical to $\mathcal{A}$ for all V_i, $i \leq l$ and hence different from $\mathcal{A}'$ on a number of locations which, by the previous paragraph and the choice of l, totals at least $\frac{m-o(m)}{3} \cdot \frac{n}{C}$. A similar argument goes for the distance from $\mathcal{A}$.

Finally, let $\mathcal{C}$ be the closest input to $\mathcal{B}$ such that $\mathcal{C}$ satisfies $\mathcal{P}_{\mathcal{G}}$. Assume first that $\mathcal{C}$ is identical to $\mathcal{A}$ on V_i for some $i \leq l$. Then $\mathcal{C}$ must be identical to $\mathcal{A}$ on every $j \geq i$. This implies that it differs from $\mathcal{A}'$, and hence from $\mathcal{B}$, in at least n/C places for every V_i, $i > l$, as was shown by the first paragraph. This totals to a distance of at least $\frac{mn(1-o(1))}{3C}$.

Assume then that there is no $i \leq l$ for which $\mathcal{C}$ is identical to $\mathcal{A}$ on V_i. Let i be the smallest index for which $\mathcal{C}$ differs from $\mathcal{A}$ in at most $n/(10C)$ places in V_i. If $i \geq \frac{10m}{33}$, then clearly $\mathcal{C}$ differs from $\mathcal{A}$, and hence from $\mathcal{B}$, in at least $n/(10C)$ places for every $j \leq i$. This would total a distance between $\mathcal{C}$ and $\mathcal{B}$ of at least $\frac{n}{10C} \cdot \frac{10m}{33} = \frac{nm}{33C}$, which proves the claim. Assume then that $i < \frac{10m}{33}$ and let S_i be the set of places in V_i where $\mathcal{C}$ differs from $\mathcal{A}$. By our assumption on i, $|S_i| \leq n/(10C)$, and hence by Item 2 in Definition 5.6, $S_i \in \mathcal{F}_i$. Now, $\mathcal{C}$ differs from $\mathcal{A}$ in the propagation set of S_i into V_j for every $j \geq i$. Note also that $S_i \in \mathcal{F}_i$ implies that its propagation set into V_j is in $\mathcal{F}_j$ for every $j \geq i$ (Item 2 of Definition 5.6). Hence, by the odd-expansion property for every G_j, this propagation set contains at least n/C elements for every $j \geq i + \Theta(\log n)$, and hence $\mathcal{C}$ differs from $\mathcal{A}$, and hence also from $\mathcal{B}$, in at least $\frac{n}{C}$ places in every level $j > i + \Theta(\log n)$. By the definition of i this totals at least $\frac{n}{C} \cdot (\frac{m}{3} - \frac{10m(1-o(1))}{33}) = \frac{n}{C}\frac{m-o(m)}{33}$ places, which completes the proof of the claim. ∎

Let $q = o(\sqrt{m})$. We now analyze the values that A and N induce on a fixed set of q nodes; let us denote the set of vertices by $S = \{u_1, \ldots, u_q\}$, and the corresponding levels by $u_1 \in V_{i_1}, \ldots, u_q \in V_{i_q}$. Let $D(S)$ be the event that $|i_j - l| > \sqrt{m}$ for every j.

Observation 5.9 *For both A and N,* $\mathrm{Prob}(D(S)) \geq 1 - o(1)$.

Proof: For every element in S there are $2\sqrt{m}$ choices of l that violate $D(S)$. As there are $q = o(\sqrt{m})$ elements in S the observation follows. ∎

We shall prove in the following that conditioned on $D(S)$, the restrictions of A and N on $u_1, \ldots, u_q$ yield the same distribution.

In order to understand the restriction of A to a set S, we use the following lemma.

Lemma 5.10 *For any set of vertices T, and any set $T' \subset T$ such that $\{ch(y) \mid y \in T'\}$ is a maximal linearly independent subset of the set of vectors $\{ch(y) \mid y \in T\}$, the following occurs.*

The restriction of the distribution A to T' is the uniform distribution on a set of $|T'|$ independent binary variables, and the values chosen for the variables associated with T' completely determine the values chosen for all of T.

Proof of Lemma 5.10: Fix a set $T' \subseteq T$ such that $\{ch(y) \mid y \in T'\}$ is a maximal linearly independent set of vectors among $\{ch(y) \mid y \in T\}$. By Observation 5.3 the value of every y is completely determined by the values of the vertices in $BP(y)$, and is in fact their parity. For a set $T'' \subseteq T'$ the probability that $\bigoplus_{y \in T''} x_y = 1$ is $\frac{1}{2}$ because the backpropagation of T'' into the first layer is nonempty. This means that the distribution over T' is uniform and independent, by Lemma 3.3.

On the other hand, because of the maximality of T', for every $y' \in T - T'$ there exists $T'' \subset T'$ such that $ch(y') = \bigoplus_{y \in T''} ch(y)$, so the value of y' is determined by the values of the vertices in T'. ∎

For the set S we now define S_u as the subset of S that contains all vertices that belong to levels above V_l (l was chosen while choosing an input according to A or N), and S_d as the set of all vertices that belong to V_l or below. We also let S'_u be a maximal subset of S_u for which $\{ch(y) \mid y \in S'_u\}$ is independent, and S'_d be a maximal subset of S_d for which $\{ch(y) \mid y \in S'_d\}$ is independent.

The following observation will be used in proving that conditioned on $D(S)$ happening (which occurs with high probability), both A and N induce the same distribution on S, assuming that each G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding.

Observation 5.11 *Given that $D(S)$ happens, for every nonempty subset S''_u of S'_u and nonempty subset S''_d of S'_d, the backpropagation of $S''_u \cup S''_d$ into V_1 is non empty.*

Proof: Let i be such that $S''_u \cap (\bigcup_{j \geq i} V_j) \neq \emptyset$, and consider the backpropagation of $S''_u \cap (\bigcup_{j \geq i} V_j)$ into V_i, denoted by $V_i(S''_u)$. Since $|S| \leq q = o(\sqrt{m})$, it holds that $V_i(S''_u) \in \mathcal{F}_i$ (by Item 1 in Definition 5.6). Hence by the fact that G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding and assuming that $D(S)$ holds, it follows that the backpropagation of S''_u into V_l has size $|V_l(S''_u)| = \Theta(n)$. Also, by Item 1 in Definition 5.6, $V_l(S''_u) \in \mathcal{F}_l$. Hence, this implies that the backpropagation of $(S''_u \cup S''_d) \cap (\bigcup_{j \geq i} V_j)$ into i for every $1 \leq i \leq l$ is non empty (and is in fact of size $\Theta(n)$), by the fact that G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding. In particular this holds for $i = 1$. ∎

And now for the final claims.

Claim 5.12 *Given that $D(S)$ happened, the distribution induced by A on $S'_u \cup S'_d$ is the uniform independent one, and the values of the members of $S - S'_u \cup S'_d$ are determined by those of $S'_u \cup S'_d$.*

Note that the claim is very similar to Lemma 5.10. The only difference is that while each of S'_u, S'_d is such that its corresponding vectors are independent, it is not a priory guaranteed that for the union $S'_u \cup S'_d$ the set of its corresponding vectors is also independent.

Proof: It is enough to prove that the set $A = \{ch(y) \mid y \in S'_u \cup S'_d\}$ is a maximal set of independent vectors, so that Lemma 5.10 will imply the claim. Hence, it is enough to prove independence, as maximality is obvious by the assumption on S'_d and S'_u.

To prove independence it is enough to show that for every two subsets $S''_u \subseteq S'_u$ and $S''_d \subseteq S'_d$, $\sum_{y \in S''_u \cup S''_d} ch(y)$ is not the zero vector. By Observation 5.5, $\sum_{y \in S''_u \cup S''_d} ch(y) = ch(BP(S''_u \cup S''_d))$, and by Observation 5.11 $BP(S''_u \cup S''_d)) \neq \emptyset$, which implies the claim. ∎

Claim 5.13 *Given that $D(S)$ happened, the distribution induced by N on $S'_u \cup S'_d$ is the uniform independent one, and the values of the members of $S - S'_u \cup S'_d$ are determined by those of $S'_u \cup S'_d$ in the same manner as with the distribution A.*

Proof: Recall that to obtain an input according to N we choose an input $\mathcal{A}$ according to A and a vertex $z \in V_1$. We then set the input $\mathcal{A}'$ (which is also in A) by flipping the value of z and take the extension of the input values to each layer. Then the input from N is identical to $\mathcal{A}$ at all levels up to l and to $\mathcal{A}'$ from level l and above. To prove the claim we use again Lemma 3.3.

Let $x_v, v \in V$ be the values of the input according to N. For every $S''_u \subseteq S'_u$ and $S''_d \subseteq S'_d$ we will show that if $S''_u \cup S''_d \neq \emptyset$ then $\bigoplus_{v \in S''_u \cup S''_d} x_v = 1$ with probability 1/2. Clearly it is enough to show this with the assumption that both S''_u and S''_d are non empty (as otherwise, this reduces to the claim for A, which was already proven in Claim 5.12).

Let A_v be the value of v according to $\mathcal{A}$ and let A'_v be the value of v according to $\mathcal{A}'$. Then $x_v = A(v)$ for $v \in V_i$ and $i \leq l$, and $x_v = A'(v)$ for $v \in V_i$ and $i > l$.

By Observation 5.3,

$$\begin{aligned}
\bigoplus_{v \in S''_u \cup S''_d} x_v &= \left(\bigoplus_{v \in S''_u} \bigoplus_{u \in BP(v)} A'(u) \right) \oplus \left(\bigoplus_{v \in S''_d} \bigoplus_{u \in BP(v)} A(u) \right) \\
&= \left(\bigoplus_{v \in S''_u} \bigoplus_{u \in BP(v)} A(u) \right) \oplus \left(\bigoplus_{v \in S''_d} \bigoplus_{u \in BP(v)} A(u) \right) \oplus \left(\bigoplus_{v \in S''_u \text{ s.t. } z \in BP(v)} 1 \right) \\
&= \left(\bigoplus_{v \in S''_u \cup S''_d} A(v) \right) \oplus \left(\bigoplus_{v \in S''_u \text{ s.t. } z \in BP(v)} 1 \right)
\end{aligned}$$

By Observation 5.4, the above is equal to $[\bigoplus_{v\in BP(S''_u\cup S''_d)} x_v]\oplus b$, where $b=\bigoplus_{v\in S''_u \text{ s.t. } z\in BP(v)} 1$ is independent of the input values.

Now, under the assumption that $D(S)$ holds, Observation 5.11 asserts that $BP(S''_u\cup S''_d)\neq\emptyset$ and hence the parity of the values in it will be 1 with probability 1/2 as required.

The values of the members of $S-S'_u\cup S'_d$ have the same dependencies that they have under the distribution A, because none of these dependencies involves members of both S_u and S_d (again by Observation 5.11). ■

Proof of Proposition 5.7: We let $\mathcal{G}$, q, A and N be as before, and prove that even an adaptive algorithm that queries at most q values $x_u, u\in V$ has an error probability of at least 1/6 on inputs taken from the distribution $D=0.5A+0.5N$ (by standard amplification this also shows that any algorithm that uses at most $q/3$ queries must have error at least 1/3).

Let $Q=\{u_1,\ldots,u_q\}\subseteq V$ be a set of vertices in the corresponding levels $u_1\in V_{i_1},\ldots,u_q\in V_{i_q}$. We denote by $D(Q)$ the event that $|i_j-l|>\sqrt{m}$ for every j.

Now let $\mathcal{T}$ be an adaptive algorithm that queries at most q values $x_u, u\in V$. Every leaf in the decision tree that represents $\mathcal{T}$ is labeled by either 'accept' or 'reject'. Let L be the set of all leaves that are labeled by 'accept'. We may assume that $\text{Prob}_P(L)\geq\frac{2}{3}$, as otherwise the algorithm errs on positive inputs with probability at least $\frac{1}{2}-\text{Prob}_D(L)\geq 1/6$. We shall prove that this necessarily implies that $\text{Prob}_N(L)>\frac{2}{3}\cdot(1-o(1))>1/3$, which implies that the algorithm errs by accepting wrong inputs with probability at least 1/6.

Every leaf $\alpha\in L$ is associated with the set of vertices $Q(\alpha)$ that were queried along the way to α and with a set of answers $ans(\alpha):Q(\alpha)\longrightarrow\{0,1\}$ to those queries. Clearly, $Q(\alpha)$ and $ans(\alpha)$ together uniquely define α. Moreover, the algorithm will reach α if and only if the variables corresponding to the nodes of $Q(\alpha)$ are respectively assigned $ans(\alpha)$, regardless of the assignments made to other variables that are queried in other branches of the decision tree of $\mathcal{T}$.

Let $V'(\alpha)\subset Q(\alpha)$ be such that $\{ch(y)\mid y\in V'\}$ is a maximal linearly independent subset of the set of vectors $\{ch(y)\mid y\in Q(\alpha)\}$. Lemma 5.10 asserts that for positive inputs the answers on $V'(\alpha)$ uniquely determine the answers on $Q(\alpha)-V'(\alpha)$. We may assume that for every $\alpha\in L$ the answers to $Q(\alpha)-V'(\alpha)$ are consistent with the answers on $V'(\alpha)$, as otherwise the leaf α may be changed to 'reject' without reducing the success probability of $\mathcal{T}$ under either A or N. Let $dim(\alpha)$ denote the size of $V'(\alpha)$. The above discussion implies the following.

Claim 5.14 *For every $\alpha\in L$,* $\text{Prob}_P(\alpha)=2^{-dim(\alpha)}$.

Proof: Immediate from Lemma 5.10. ■

Claim 5.15 *Every $\alpha\in L$ satisfies that* $\text{Prob}_N(\alpha)\geq(1-o(1))2^{-dim(\alpha)}=(1-o(1))\text{Prob}_P(\alpha)$

Proof: Note that $\text{Prob}_N(\alpha)\geq\text{Prob}_N(\alpha\wedge D(Q(\alpha)))=\text{Prob}_N(D(Q(\alpha)))\cdot\text{Prob}_N(\alpha|D(Q(\alpha)))\geq(1-o(1))\cdot 2^{-dim(\alpha)}$. The last inequality follows from Observation 5.9 together with the fact that $\text{Prob}_N(\alpha|D(Q(\alpha)))=2^{-dim(\alpha)}$, which follows from Claim 5.13. ■

Claim 5.15 implies that $\text{Prob}_N(L) = \sum_{\alpha \in L} \text{Prob}_N(\alpha) \geq (1-o(1)) \cdot \sum_{\alpha\in L} 2^{-dim(\alpha)} = (1-o(1)) \cdot \text{Prob}_P(L) \geq (1-o(1)) \cdot 2/3$, which completes the proof of Proposition 5.7. ■

5.2 The existence of an odd-expanding circuit

We now prove that there exists $\mathcal{G} = \{G_1, \ldots, G_{m-1}\}$ such that every G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding for $\mathcal{F}_i$ as per Definition 5.6, which concludes the proof of Theorem 5.1 (by satisfying the conditions of Proposition 5.7) and hence of the lower bound. First we need the following lemma.

Lemma 5.16 *Let $(4.5^{2.5} \cdot e^{12.5})^{(2/3)} < 51000 \leq C$, let $|U| = |V| = n$, and suppose that $\mathcal{F}$ is a family of at most $e^{n/C}$ nonempty subsets of U. Then $G(U \cup V, E)$, where E is the union of 7 random perfect matchings between U and V, is $(\frac{1}{C}, \mathcal{F})$-odd-expanding with probability at least $(1 - O(n^{-3/2}))$.*

Note that G may have parallel edges with constant probability.

Proof of Lemma 5.16: To prove the odd-expansion properties with regards to a fixed family $\mathcal{F}$ we analyze the odd-expansion of subsets $S \subseteq U$ according to their sizes. We first show that with very high probability G will be odd-expanding for all small enough sets, using the standard expansion property of random graphs. Only for larger sets we need the bound on the size of $\mathcal{F}$.

For a set S of size k, $k \leq n/C$, let us look at the 7 random sets that are the images of S under the 7 random matchings. Each such image $M_i(S)$ is a random set of size k and is independent of the others. We first want to bound the probability that the union of these sets is small: Let S be fixed and let $T \subseteq V$ be of size $4.5k$. Using the inequalities $(\frac{n}{k})^k \leq \binom{n}{k} \leq (\frac{ne}{k})^k$ we get:

$$\text{Prob}\left[\bigcup_{i=1}^{7} M_i(S) \subseteq T\right] \leq \frac{\binom{4.5k}{k}^7}{\binom{n}{k}^7} \leq \frac{(4.5e)^{7k}}{(\frac{n}{k})^{7k}} = \left(\frac{4.5ek}{n}\right)^{7k}.$$

Hence, by summing for all possible $k \leq n/C$ and for all possible sets S and T we get:

$$\text{Prob}\left[\exists S, |S| = k \leq n/C, \left|\bigcup_{i=1}^{7} M_i(S)\right| \leq 4.5k\right] \leq \sum_{k=1}^{n/C} \binom{n}{k} \cdot \binom{n}{4.5k} \cdot \left(\frac{4.5ek}{n}\right)^{7k} \leq$$

$$\sum_{k=1}^{n/C} \left(\frac{ne}{k}\right)^k \cdot \left(\frac{ne}{4.5k}\right)^{4.5k} \cdot \left(\frac{4.5ek}{n}\right)^{7k} = \sum_{k=1}^{n/C} \left(\frac{4.5^{2.5} \cdot e^{12.5} \cdot k^{1.5}}{n^{1.5}}\right)^k \leq$$

$$\frac{4.5^{2.5} \cdot e^{12.5}}{n^{1.5}} + \sum_{k=2}^{\frac{n}{2eC}} \left(\frac{4.5^{2.5} \cdot e^{12.5} \cdot k^{1.5}}{n^{1.5}}\right)^k + \sum_{k=\frac{n}{2eC}}^{n/C} \left(\frac{4.5^{2.5} \cdot e^{12.5} \cdot k^{1.5}}{n^{1.5}}\right)^k \leq$$

$$\frac{4.5^{2.5} \cdot e^{12.5}}{n^{1.5}} + \sum_{k=2}^{\frac{n}{2eC}} \frac{4.5^{2.5} \cdot e^{12.5} \cdot 8}{n^{1.5}} \cdot 2^{-k+2} + n \cdot \left(\frac{4.5^{2.5} \cdot e^{12.5}}{C^{1.5}}\right)^{n/2eC} = O(n^{-3/2}).$$

The last bound on the summands is by our choice of C. It guarantees that the first series is dominated by a geometric series while in the second series each summand is at most $(\frac{4.5^{2.5} \cdot e^{12.5}}{C^{1.5}})^{n/2eC}$, which is exponentially small in n.

Now note that if $|N(S)| \geq 4.5|S|$ then $|O_G(S)| \geq 2|S|$. To see this we define $N^1(S)$ to contain all vertices $v \in N(S)$ which have exactly one neighbor in S. Clearly $N^1(S) \leq O_G(S)$. However, $|E(G[S, N(S)])| = 7|S| \geq |N^1(S)| + 2|N(S) - N^1(S)|$. This, together with $|N(S)| \geq 4.5|S|$, implies that $|O_G(S)| \geq |N^1(S)| \geq 2|S|$, which is more than what we need.

Now let $S \subseteq V$ be of size k, $n/C \leq k \leq n/2$. Fix $i \in \{1, \ldots, 7\}$ and let $T_j = V - M_j(S)$ for $j \neq i$. Let $P_i = M_i(S) \cap (\bigcap_{j \neq i} T_j)$, namely P_i contains all elements that are in the i'th image of S but not in any $M_j(S), j \neq i$. Certainly $\bigcup_{i=1}^{7} P_i \subseteq O_G(S)$.

To estimate the size of $\bigcup_i P_i$, we simulate the process of choosing a random matching of S by marking each element of V independently, with probability k/n. We then repeat it independently 7 times. Finally, we condition this on the event that $M_i(S)$, the set marked in the i'th round, is of size exactly k for every i, and pick a random 1-1 mapping between S and each of the 7 marked sets. Let B_i be the event that $|M_i(S)| = k$. Then $\text{Prob}(B_i) \geq \Omega(1/\sqrt{n})$ (since $|M_i(S)|$ is binomially distributed, this follows using the Chebyshev inequality [1]). Note also that in the random model where we do not condition on the events $B_r, r = 1, \ldots, 7$, the expected size of $\bigcup_i P_i$ is $7k \cdot (1 - \frac{k}{n})^6 \geq \frac{6.9n}{C}$ by our choice of k. As this is a sum of independent random Boolean variables we may apply a Chernoff bound on the probability of the deviation from the expectation being large. Hence we get (for sufficiently large n):

$$\text{Prob}\left[\left|\bigcup_{i=1}^{7} P_i\right| \leq \frac{n}{C} \mid \bigwedge_{r=1}^{7} B_r\right] \leq \text{Prob}\left[\left|\bigcup_{i=1}^{7} P_i\right| \leq \frac{n}{C}\right] \cdot \left(\text{Prob}\left[\bigwedge_{r=1}^{7} B_r\right]\right)^{-1}$$
$$\leq exp\left(-\frac{2.48n}{C}\right) \cdot n^{3.5} \leq \frac{1}{n^2} exp\left(-\frac{2.4n}{C}\right)$$

Hence with probability of at least $1 - \frac{1}{n^2} \cdot e^{-\frac{2.4n}{C}}$ the union of the P_i's has size at least $\frac{n}{C}$, which implies that $|O_G(S)| \geq \frac{n}{C}$ as required. Thus for any fixed family $\mathcal{F}$ of at most $e^{2.4n/C}$ sets of sizes of at most $n/2$ we have that G is $(1/C, \mathcal{F})$-odd-expanding with probability at least $1 - n^{-2}$.

Finally, for larger sets: Let $S \subseteq V$ with $|S| = k \geq n/2$. Let T be the set of all elements of V that are the images of S for all M_i, namely $T = \cap M_i(S)$. Certainly $T \subseteq O_G(S)$. A similar analysis to the above shows that the expected value of $|T|$ is $n(\frac{k}{n})^7 \geq \frac{n}{128}$. Hence with probability at most $exp(-\frac{9}{16} \cdot \frac{n}{2 \cdot 128}) \leq exp(-n/C)$ we may get $|T| \leq n/C$. Again this makes sure that with probability $1 - n^{-2}$ the graph G is $(1/C, \mathcal{F})$-odd-expanding for every fixed $\mathcal{F}$ containing at most $2^{n/C}$ sets of sizes above $n/2$ each. This, together with the previous cases, completes the proof of the lemma (the argument for subsets of V is identical to that for subsets of U). ∎

Lemma 5.17 *Suppose that $\mathcal{G} = G_1, \ldots, G_{m-1}$ is a family of graphs in which each G_i is constructed randomly and independently as was done in Lemma 5.16. Suppose also that for every i, $\mathcal{F}_i$ is defined as in Definition 5.6. Then with a positive probability every G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding.*

Proof: For every $1 \leq i \leq m$ let H_i be the event that G_i is $(\frac{1}{C}, \mathcal{F}_i)$-odd-expanding. Note that $\mathcal{F}_i$ is dependent only on $\{G_j \mid j \neq i\}$ and not on G_i itself, which is chosen independently of them. Also note that $|\mathcal{F}_i| \leq e^{n/C}$. Thus Lemma 5.16 bounds from below the probability of H_i happening by $1 - O(n^{-3/2})$. In particular, with positive probability all the events H_i occur. ∎

6 Concluding comments

- It would be interesting to prove a constructive version of Lemma 5.17, thus reducing the uniform computational complexity class of our second property.
- Using the notation from [18], where a BP is allowed to reject an input before its calculation is complete, our property has a width-two read-twice BP, while BP's which are fixed-width read-once, and BP's which are width-one, were shown in [18] to decide testable languages only.
- It would be interesting to formulate restrictions on the order in which the input is read, that assure testability of the language even if the BP is not read-once. For example, one can see that certain orderings (e.g., reading the input $x = x_1, \ldots, x_n$ in its natural order twice) assure testability also for read-twice BP's.
- It would be interesting to know for which values of k and l there exists $\alpha < 1$, such that the number of queries required to test any language recognizable by an oblivious width-k read-l-times BP is bounded by n^{α}.

Acknowledgment

We wish to thank an anonymous referee for suggestions and comments.

References

[1] N. Alon and J. H. Spencer, The probabilistic method, *John Wiley & Sons, Inc.* 1991.

[2] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy, Efficient testing of large graphs, *Combinatorica* 20:451–476, 2000.

[3] N. Alon, M. Krivelevich, I. Newman and M. Szegedy, Regular Languages are Testable with a Constant Number of Queries, *SIAM J. of Computing* 30(6): pp. 1842-1862 (2000).

[4] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof verification and the hardness of approximation problems, *JACM*, 45(3):501–555, 1998.

[5] T. Batu, R. Rubinfeld, P. White, Fast Approximate PCPs for Multidimensional Bin-Packing Problems, In *3rd RANDOM-APPROX99 Conference Proceedings*, pp. 245–256, Springer-Verlag, Berkeley CA, 1999.

[6] E. Ben-Sasson, P. Harsha and S. Raskhodnikova, Some 3CNF properties are hard to test, In 35^{th} *ACM STOC Conference proceedings*, pp. 345–354, 2003.

[7] M. Blum, M. Luby and R. Rubinfeld, Self-testing/correcting with applications to numerical problems. *JCSS*, 47:549–595, 1994.

[8] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron and A. Samorodnitsky, Improved testing algorithms for monotonicity, In 3^{rd} *RANDOM Conference Proceedings*, pp. 97–108, Springer-Verlag, Berkeley CA, 1999.

[9] E. Fischer, On the strength of comparisons in property testing, *Information and Computation*, in press.

[10] E. Fischer, Testing graphs for colorability properties, In 12^{th} *SODA Conference Proceedings*, pp. 873–882, 2001.

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

[12] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld and A. Samorodnitsky, Monotonicity testing over general poset domains, In 23^{th} *ACM STOC Conference Proceedings*, pp. 474-483, 2002

[13] E. Fischer and I. Newman, Functions that have read-twice constant width branching programs are not necessarily testable, In 17^{th} *Conference on Computational Complexity Proceedings*, pp. 73–79, 2002.

[14] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan and A. Wigderson, Self-testing/correcting for polynomials and for approximate functions, In 23^{th} *ACM STOC Conference Proceedings*, pp. 32–42, 1991.

[15] O. Goldreich, S. Goldwasser and D. Ron, Property testing and its connections to learning and approximation, *JACM*, 45(4):653–750, 1998.

[16] O. Goldreich and D. Ron, Property testing in bounded degree graphs, *Algorithmica*, 32(2):302–343, 2002.

[17] J. W. P. Hirschfeld, Projective geometries over finite fields, *The Clarendon Press, Oxford University Press*, 1979/1998.

[18] I. Newman, Testing of Function that have small width Branching Programs, *SIAM Journal on Computing*, 31:1557–1570, 2002.

[19] D. Ron, Property testing (a tutorial), In: *Handbook of Randomized Computing* (S. Rajasekaran, P. M. Pardalos, J. H. Reif and J. D. P. Rolim eds), Kluwer Press, 2001.

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

[21] J. H. Van Lint and R. M. Wilson, A course in combinatorics, *Cambridge University Press*, 1992/2001.

[22] A. C. Yao, Probabilistic computation, towards a unified measure of complexity, In 18^{th} *IEEE FOCS Conference Proceedings*, pp. 222–227, 1977.