Testing of matrix-poset properties*

Eldar Fischer† Ilan Newman‡

February 7, 2007

Abstract

Combinatorial property testing, initiated by Rubinfeld and Sudan [23] and formally defined by Goldreich, Goldwasser and Ron in [18], deals with the following relaxation of decision problems: Given a fixed property P and an input f, distinguish between the case that f satisfies P, and the case that no input that differs from f in less than some fixed fraction of the places satisfies P. An (ϵ, q)-test for P is a randomized algorithm that queries at most q places of an input f and distinguishes with probability $2/3$ between the case that f has the property and the case that at least an ϵ-fraction of the places of f need to be changed in order for it to have the property.

Here we concentrate on labeled, d-dimensional grids, where the grid is viewed as a partially ordered set (poset) in the standard way (i.e. as a product order of total orders). The main result here presents an $(\epsilon, poly(1/\epsilon))$-test for every property of 0/1 labeled, d-dimensional grids that is characterized by a finite collection of forbidden induced posets. Such properties include the 'monotonicity' property studied in [9, 8, 13], other more complicated forbidden chain patterns, and general forbidden poset patterns. We also present a (less efficient) test for such properties of labeled grids with larger fixed size alphabets. All the above tests have in addition a 1-sided error probability. This class of properties is related to properties that are defined by certain first order formulae with no quantifier alternation over the syntax containing the grid order relations.

We also show that with one quantifier alternation, a certain property can be defined, for which no test with query complexity of $O(n^{1/4})$ (for a small enough fixed ϵ) exists. The above results identify new classes of properties that are defined by means of restricted logics, and that are efficiently testable. They also lay out a platform that bridges some previous results.

AMS Subject classification: 94A20, 68Q99, 68R05

*A preliminary version of these results formed part of [14]. This research was funded in part by the Israel Science Foundation, grant number 55/03

†Department of computer science, The Technion, Haifa 32000, Israel. Email: eldar@cs.technion.ac.il

‡Department of Computer Science, University of Haifa, Haifa, Israel. Email: ilan@cs.haifa.ac.il

1 Introduction

Combinatorial property testing deals with the following relaxation of decision problems: Given a fixed property and an input f, one wants to decide using as few queries to f as possible whether f has the property or is 'far' from having the property. The general notion of property testing was first formulated by Rubinfeld and Sudan [23], who were motivated mainly by its connection to the study of program checking. The study of this notion for combinatorial objects, and mainly for labeled graphs, was introduced by Goldreich, Goldwasser and Ron [18]. A property in this respect is an infinite language, where each member is a Boolean functions from a set (usually with some structure) to $\{0,1\}$ (sometimes a larger range is considered). Being far is measured by the *hamming distance*, namely, in how many places should the input function be changed so as to have the property. An input function here is identified with its table, namely its $0/1$ value for each of the points of the domain. A property is said to be $(\epsilon, q(\epsilon, n))$-*testable* if there is a randomized algorithm that for every input function f over a domain of size n, queries the values of f on at most $q(\epsilon, n)$ chosen points of the domain, and with probability $\frac{2}{3}$ distinguishes between the case that f has the property and the case that f is ϵn-far from having the property. When a property P is (ϵ, q)-testable with $q = q(\epsilon)$ (i.e. q is a function of ϵ only, and is independent of n) then we say that P is ϵ-*testable*; we say that P is *testable* if it is ϵ-testable for every $\epsilon > 0$. 1-*sided testability* is defined in the same manner, but with the additional requirement that the algorithm accepts an input that satisfies P with probability 1.

Property testing has recently become quite an active research area, see e.g. [18, 19, 8, 5, 1, 3, 21, 11] for an incomplete list, and also the surveys [22, 12]. Apart from its theoretical appeal, and the many questions it involves, it emerges in the context of PAC learning, program checking [16, 6, 23], probabilistically checkable proofs [4] and approximation algorithms [18]. The advantage of the 'property testing' relaxation is that many properties have a randomized test that reads a very short piece of the input and also runs very fast (in sublinear time).

One of the main tasks that emerged in the field, following [18], is to identify natural collections of properties that are efficiently testable (in terms of the number of queries). Goldreich et al. [18] studied some classes of properties (mainly graph properties) and identified many properties that are testable. Alon et al. [3] considered properties of functions $f : \{1, \ldots, n\} \longrightarrow \{0, 1\}$, namely, where each function is a binary string of length n. They suggested that properties that are defined by restricted logics might be testable. They proved that every regular language is testable, which is equivalent to saying that every property that is expressible by a certain second order monadic logic over ordered sequences is testable. Additional work in this direction was done by [21], generalizing the above, and by [1, 11] on graph properties. In [9, 17, 8, 10], the specific property of 'monotonicity' is studied.

The results in [3, 9, 10] deal with properties whose definition relies on a linear ordering of the domain of the input, and in [17, 8, 13] the definition of the monotonicity property relies on other partial orderings. Here we make another step in the direction established above: We present a logical model (and discuss some variants) such that all properties that can be expressed in it are testable. Our structure is the d-dimensional grid $\{1, \ldots, n\}^d$, equipped with the natural product order. The logical model we use is that of first order formulae with the order relation. Our main positive result is that for every fixed d, every such formula that uses no quantifier alternation (i.e. using only one quantifier on a fixed sequence of points) is testable. 'Non trivial' such properties (this will be formally defined in the next section) have another equivalent combinatorial formulation: Every such property is characterized by a finite collection of forbidden induced sub-posets (and the converse also holds). Such properties include 'monotonicity' (stating that there are no two points x, y, $x \leq y$, that are labeled $1, 0$ respectively). The model also includes more complicated properties of chains, e.g: "All 2-dimensional, 0/1-labeled grids that when going towards the north-east direction in any possible way, contain no sequence of 4 points A,B,C,D labeled 0,1,0,1". More complicated examples are, e.g: "All 2-dimensional, 0/1-labeled grids that contain no four points A,B,C,D labeled by, say, 1,1,0,1 respectively, and such that A is south-west (SW) of all, D is NE (north-east) of all, and neither B is SW of C, nor C is SW of B". We present a 1-sided error $(\epsilon, poly(1/\epsilon))$-test for such properties for every fixed dimension d. We also consider matrices over an alphabet (i.e. set of possible labels) which is not 0/1, but any fixed finite set. For these we obtain a less efficient 1-sided test, that makes $exp(exp(poly(1/\epsilon)))$ many queries.

On the negative side, we show that there exists a poset (matrix) property expressed by using only one "$\forall\exists$" quantifier alternation which is not ϵ-testable for some ϵ. This puts a bound on the complexity of the relevant logical model that still guarantees testability. To put the results in perspective, we note that our model for the one dimensional case includes only regular properties, so our result for the case of 1-dimensional "grids" is a special case of the results of [3]. We also note that an instance of the matrix properties that we discuss here (for all d) is the extensively studied property of 'monotonicity', namely the property: $\forall x_1, x_2\, ((x_1 \leq x_2) \rightarrow (M(x_2) \vee \neg M(x_1)))$, [9, 17, 8] ('$M(x)$' means 'the value of the input matrix at x is 1').

The rest of the paper is organized as follows: In Section 2 we define the basic model and prove some basic preliminaries, in Section 3 we construct as a warm-up a simple to prove, 2-sided error, ϵ-test, and then construct the full 1-sided error test for the basic model of properties (one quantifier, binary matrices). In Section 4 we construct the test for properties of matrices over any fixed, non-binary, alphabet. In Section 5 we construct a property that can be expressed as a formula with one quantifier alternation and which is not testable, by starting with a property that belongs to a more general model of matrix properties and deriving our property from it. Finally, in Section 6 we discuss variants of the poset-models, and pose some open problems and future directions.

2 Some preliminaries and notations

In the following, we omit all floor and ceiling signs whenever the implicit assumption that a quantity is an integer makes no essential difference. We make no attempt to optimize the coefficients involved, just the function types (e.g. polynomial versus exponential).

Let $[n] = \{0, \ldots, n-1\}$ with the natural order "$\leq$". The *n-length d-dimensional grid* is the poset (partially ordered set) $G(n, d) = [n]^d$ with the product order, namely the order defined by stating that $(\alpha_1, \ldots, \alpha_d) \leq (\beta_1, \ldots, \beta_d)$ if $\alpha_i \leq \beta_i$ for all $i = 1, \ldots, d$.

For $\alpha = (\alpha_1, \ldots, \alpha_d)$ and $\beta = (\beta_1, \ldots, \beta_d)$, if $\alpha_j = \beta_j$ for some $1 \leq j \leq d$, then we say that α and β *share a coordinate*.

A Boolean function $f : G(n, d) \longrightarrow \{0, 1\}$ is identified with a copy of $G(n, d)$ whose points are labeled by 0/1. Such a function will be called a 0/1 (n, d)-*matrix*, or just an (n, d)-matrix. A standard 2-dimensional 0/1, $n \times n$ matrix is by this notation an $(n, 2)$-matrix. Hence a property $\mathcal{P} = \{\mathcal{P}_n\}_{n=1}^{\infty}$, of functions over the structure $G(n, d), n \in \mathbb{N}$, is just a set of (n, d)-matrices for each $n \in \mathbb{N}$. For two (n, d)-matrices M and R, their distance is defined by $dist(M, R) = |\{x : \ M(x) \neq R(x)\}|$. For an (n, d)-matrix M and a property $\mathcal{P}$ we define $dist(M, \mathcal{P}) = \min_{R \in \mathcal{P}_n} dist(M, R)$.

In the basic logical model that we study, the variables range over $G(n, d)$. The syntax includes the poset (binary) relation and the function unary relation $M(\cdot)$ ("being labeled 1"). Given a fixed set of variables, $x_1, \dots, x_k$, a Boolean formula $\phi(x_1, \dots, x_k)$ using the above relations specifies an allowed set of 0/1-labeled posets that are its truth assignments. The basic model of properties of (n, d)-matrices contains the properties that can be expressed as $\forall x_1, \dots, x_k \phi(x_1, \dots, x_k)$ where ϕ is a formula as above and k is a fixed constant (independent of n). For example, the well studied property 'monotonicity' is such: $\forall x_1, x_2 \, ((x_1 \leq x_2) \to (M(x_2) \vee \neg M(x_1)))$. We call this model the $\forall$-*poset model*. Similarly the $\exists$-poset, the $\forall\exists$-poset models, etc. are defined.

Clearly if a property is expressible as an $\exists\phi(x_1, \dots, x_k)$ formula then either every matrix has distance at most k to it, or the property is empty. Hence, such properties are trivially ϵ-testable. Things start to be more complex for $\forall$ properties. Our main result is that any such property is testable for $d = O(1)$. We then show that there exists a $\forall\exists$-poset property that is not testable even for dimension $d = 2$.

We first note that testing a $\forall$-poset property is equivalent to testing that there is no forbidden fixed substructure. To make this explicit we need some definitions and observations: We say that a poset P of size k is a *subgrid poset of dimension* d if there is a set of k points in $G(n, d)$ (for some n) on which the induced order is isomorphic to P. We say that P is a 0/1-labeled poset if every point of P is labeled by 0 or 1. An (n, d)-matrix M contains a labeled poset P of size k as an *induced labeled poset* if there are k points in M on which the order relation is isomorphic to P and the point labels are consistent with the labeling of P.

Definition 2.1 *Let F be a set of 0/1-labeled posets. The following property of (n, d)-matrices is defined: $\mathcal{M}_F = \{M : \ M$ does not contain any member of F as an induced labeled poset $\}$.*

Observation 2.2 *Let $\mathcal{P}$ be a $\forall x_1, \dots, x_k \phi(x_1, \dots, x_k)$ type property of (n, d)-matrices. Then there exists a set F of labeled k-size posets, $|F| \leq 2^{k^2}$, such that $\mathcal{P} = \mathcal{M}_F$.*

Proof: We can rewrite the negation of $\mathcal{P}$ as $\exists x_1, \dots, x_k \neg\phi(x_1, \dots, x_k)$. In turn, we can write an equivalent DNF formula $\bigvee_i m_i(x_1, \dots, x_k)$ for $\neg\phi$, where each $m_i(x_1, \dots, x_k)$ represents a labeled poset F_i on at most k elements. Hence, a matrix satisfies $\mathcal{P}$ if and only if it has no F_i as a labeled-subgrid-poset. ∎

We also note that every property $\mathcal{M}_F$ is a $\forall x_1, \dots, x_k \phi(x_1, \dots, x_k)$ type property and therefore the above reduction is an equivalence. Along the sequel, our strategy for testing M for the property $\mathcal{M}_F$ will be to query some points in M and try to locate a member of F as an induced labeled poset within the queried points.

3 Algorithms for testing poset properties

We present here a 1-sided error test for $\mathcal{M}_F$ whose query complexity is polynomial in ϵ^{-1}. To illustrate the basic scheme and highlight the issues that occur with it, we start with the following almost trivial algorithm for the 1-dimensional case (the assertion of the following observation is mostly a special case of previous works such as [3], and is proven here only for imparting some basic ideas).

Observation 3.1 *For every string (1-dimensional matrix) s of length k, there exists an ϵ-test of strings for the property of not containing s as a substring, which makes a number of queries that is polynomial in ϵ^{-1}.*

Proof: Given an input string T of size n, we partition it into $m = 2k/\epsilon$ contiguous blocks. From every block we make $\Theta(\log m/\epsilon)$ many queries to detect with probability at least $\frac{1}{6m}$ whether the block contains at least an $\epsilon/2$ fraction of '1' entries, and/or at least an $\epsilon/2$ fraction of '0' entries. Let E denote the event that there was no undetected value occurring in more than an $\epsilon/2$ fraction of the entries of any block. Clearly, the probability of E is at least $\frac{2}{3}$.

If at least k blocks were detected to contain both '0' and '1' entries, the input clearly contains any string of size k as a substring and we reject it. Otherwise, we construct a string of length n (that is, a $(1, n)$-matrix), T_Q from T in the following manner. For every block of T in which no '1' was found, we make the entire block of T_Q consist of '0' entries. For every block of T in which no '0' was found, we make the entire block of T_Q consist of '1' entries. For a block that is known to have both '0' and '1' entries, the corresponding entries in T_Q will all be equal in value to those of the nearest block in which not both '0' and '1' were detected.

The string T_Q can be constructed using only the queries that were actually made to T. Also, it is not hard to see that given that E happened, and since at most k blocks contained both '0' entries and '1' entries, T_Q is ϵ-close to T. Also, for n large enough (so that every block contains at least $2k$ entries), if T_Q contains the forbidden string s then so does T. Hence, we can accept or reject T based on T_Q, and thus conclude our test. ■

The above is a scheme that is used throughout this paper: We try to obtain an approximation of the input matrix which consists mostly of "simple" blocks. In the binary setting these are blocks which are known to be almost *monochromatic*; all '0' or all '1'.

In the move to the d-dimensional case, two problems occur. First, it is no longer immediately apparent that we can immediately reject a matrix for which many blocks were detected to have both '0' and '1'. This is the easier problem, solvable by using an appropriate Zarankiewicz type lemma.

The second problem is that now we can no longer assume that it is enough to check an approximate version of our original matrix all of whose blocks are monochromatic. In Subsection 3.1, we formally present the extension of the basic ideas to the d-dimensional context, where this second problem is solved by the brute-force approach of just checking every other imaginable approximating matrix. This results in a relatively simple testing algorithm, but one that has a 2-sided error, and even worse, has a running time that is exponential in the input size despite the constant query complexity. However, the algorithm serves as a good interim result on the road towards the improved one.

To construct a 1-sided error test and reduce the running time (by avoiding the need for brute force search), we will need to consider the partition into blocks together with a refinement, and use a Ramsey-like lemma that will allow us to come back to the case where all blocks in the approximating matrix are uniform. The lemma is stated and proven in Subsection 3.2, and then in Subsection 3.3 we are finally able to present our 1-sided test for $\mathcal{M}_F$ and prove its correctness. This test will make a number of queries polynomial in ϵ^{-1} (for a fixed F) and will have a running time that is polynomial in the time it takes to make the queries.

3.1 Preliminaries and a simple 2-sided error test

Let F be a finite collection of k-size posets. We present here a basic approach to matrix property testing, along with a 2-sided error ϵ-test for $\mathcal{M}_F$, whose number of queries is polynomial in ϵ^{-1} for any fixed dimension d.

Proposition 3.2 *For every fixed $\forall$ poset property of (n, d) matrices with $d = O(1)$, there exists a 2-sided ϵ-test which makes a number of queries that is polynomial in ϵ^{-1} and is independent of the size of the input.*

The proof of Proposition 3.2 is implied by the algorithm below and the following Lemmata.

Let M be an (n, d)-matrix which we want to ϵ-test for $\mathcal{M}_F$. Let $m = (\frac{2^{k+1}k}{\epsilon^{k+1}})^{d-1} + 1$, where k is the size of the largest labeled poset in F. We divide M into m^d blocks of size $(n/m)^d$, by dividing $[n]$ into m equal-size intervals and taking Cartesian products. We now make $\frac{8d \ln m}{\epsilon}$ queries in every block of M, choosing each query uniformly at random and independently of the other queries. We tag each block as being 1, 0 or X according to the queries made: If all points that are queried in a block are labeled by '1' we tag it as '1', similarly if all points are labeled by '0' the block is tagged by '0'. Otherwise, we tag it as X. Hence we get an (m, d)-matrix M_B, in which each entry represents a block of M and is labeled by 0,1 or X.

There are two major cases: The first one is the case where at least an $(\epsilon/2)$-fraction of the blocks are tagged X. In this case we answer 'No'. It will be proven below that if this happens then there exists an actual member of F within the entries of M that were queried.

The second case is when there are less than an $(\epsilon/2)$-fraction of the blocks that are tagged X. In this case our intention is to check whether there is a matrix that is consistent with our knowledge of M as represented by M_B, and has the property. If we find such a matrix we answer 'Yes', and if not we answer 'No'. Formally, Let M_Q be the following (n, d)-matrix: For every 0-block of M, all corresponding entries of M_Q are '0', and for every 1-block of M all corresponding entries in M_Q are '1'. The entries of M_Q that correspond to an X-block of M remain undefined. Now, each possibility of assigning 0/1 values to the undefined entries of M_Q and each possible choice of flipping the values in at most an $\epsilon/4$ fraction of the entries in every other block results in a 0/1-labeled matrix; we denote the set of all such matrices by $\mathcal{M}_{Q,\epsilon}$. We check if any of the members of $\mathcal{M}_{Q,\epsilon}$ has the property. If there is such a member, the algorithm answers 'Yes'. Otherwise, if every member of $\mathcal{M}_{Q,\epsilon}$ contains a member of F, the answer is 'No'. Note that this last phase of the algorithm involves no additional queries and is just a (lengthy) computation phase.

Clearly the query complexity of this algorithm is $poly(1/\epsilon)$ and is independent of n. We now show that it is correct with high probability.

We first show that if the algorithm answers 'No' due to the fact that at least an $\epsilon/2$ fraction of the blocks are X-tagged, then it is correct with probability 1:

Claim 3.3 *If the fraction of* X*-blocks is at least $\epsilon/2$, then the queried locations in the* X*-blocks already contain a counter-example to the property.*

Proof: Let M_B be the (m,d)-matrix that was defined above. We use the following d-dimensional Zarankiewicz type theorem (see [26, 20]) that locates a (k,d)-submatrix of a given label inside any large enough matrix with enough entries of this label.

Lemma 3.4 ([20]) *For $\delta < 1$ let M be an (m,d)-matrix in which at least δm^d of the entries are marked by 'X'. If $m > (\frac{k}{\delta k+1})^{d-1}$ then there is a (k,d)-submatrix all of whose entries are 'X'.*

Now, M_B satisfies Lemma 3.4 with $\delta = \frac{\epsilon}{2}$, so there is a (k,d)-submatrix W of M_B which is all X. Our intent is to look back at the blocks of M that correspond to the entries of W. Each such block, being tagged by X, contains both a '0' and a '1' entry. We intend to argue that any labeled poset with k elements may be found within these blocks by choosing the right labeled entry. The only difficulty is that for two entries that share a coordinate in W and are comparable, the corresponding sampled entries in M might not be comparable. To overcome this difficulty we use the following lemma.

Lemma 3.5 *Any d-dimensional grid poset P with $|P| = k$ can be embedded in $G(k,d)$ with no two points sharing a coordinate.*

Proof: It is enough to prove that P can be embedded in some d-dimensional grid with no two points sharing a coordinate, since we can then take the minimum subgrid containing it and it will clearly be a (k,d)-grid.

Suppose that $f : P \to \{0,\ldots,m-1\}^d$ is any embedding of P into $G(m,d)$ for some m. Then we can think of f as a sequence of d functions $f_i : P \to \{0,\ldots,m-1\}$, $i = 1,\ldots,d$, so that $f_i(p)$ is the i-th coordinate of the point p in $G(m,d)$. We define a new embedding $f' : P \to G(m^{d+1},d)$ by the following formula (here $f(p)$ is taken as a d-dimensional row vector and $f'(p)$ is defined as a linear combination of such vectors).

$$f'(p) = m^d \cdot f(p) + \sum_{i=1}^{d} f_i(p) m^{i-1} \cdot (1,\ldots,1)$$

It is not hard to see that this is an embedding of the poset too, and that no two members share a coordinate (as the location of every member modulo m^d is unique in all coordinates). ■

To end the proof of Claim 3.3, let P be an arbitrary member of F. By Lemma 3.5, P can be embedded in W with no two points sharing a coordinate. For each point of W that corresponds to a point of P in this embedding, let us choose an entry in the corresponding block of M that has the same value (such a point exists as the points of W correspond to X-tagged blocks). These entries are an embedding of P in M. ■

We now claim that with high probability, the tagging of the blocks of M essentially represents the true situation.

Definition 3.6 *For a set Q of queries, we denote by E the following event: "Every 1-tagged block contains at most an $\epsilon/4$ fraction of '0' entries, and every 0-tagged block contains at most an $\epsilon/4$ fraction of '1' entries".*

Claim 3.7 *With probability at least $1 - 1/m^d$ the event E happens.*

Proof: Assume that a block contains at least a fraction of $\epsilon/4$ '0's. To be tagged '1', all queries in it must be '1', hence this will happen with probability at most $(1 - \frac{1}{4}\epsilon)^{\frac{8d \ln m}{\epsilon}} \leq e^{-2d \ln m} \leq m^{-2d}$. The bound for a block with at least an $\epsilon/4$ fraction of '1's is similar. Hence the probability that there is a block violating E is bounded by $1/m^d$. ■

The following claim gives us the final missing piece in the proof of the correctness of the algorithm.

Claim 3.8 *If there are less than an $\epsilon/2$ fraction of the blocks that are tagged by* X*, and the event E has happened, then the algorithm will not reject a matrix that satisfies $\mathcal{M}_F$ or accept a matrix that is ϵ-far from satisfying it.*

Proof: Assume that indeed there are less than an $\epsilon/2$ fraction of the blocks that are tagged by X, and that the event E has happened. If there exists a member of $\mathcal{M}_{Q,\epsilon}$ that has the property, then we claim that M is close to having the property. Indeed, if E has happened then M is clearly at most $\frac{1}{4}\epsilon$-far from M_Q (disregarding the entries in the X-tagged blocks), while M_Q is at most $\frac{3}{4}\epsilon$-far from any member of $\mathcal{M}_{Q,\epsilon}$ (counting *all* the entries in the X-tagged blocks), so M is at most ϵ-far from the member of $\mathcal{M}_{Q,\epsilon}$ having the property. Hence, if the algorithm answered 'Yes' then M is ϵ-close to satisfying $\mathcal{M}_F$.

On the other hand, if no member of $\mathcal{M}_{Q,\epsilon}$ has the property, then clearly M does not have the property since given that E happened, M is in particular a member of $\mathcal{M}_{Q,\epsilon}$. This completes the proof the claim. ■

Proof of Proposition 3.2 The algorithm presented above clearly has the required queried complexity. By Claim 3.3 the algorithm will never err if it rejects the input matrix on account of finding too many X-tagged blocks. In all other cases, by Claim 3.8 the algorithm may only err if the event E did not occur. However, this event occurs with probability at least $1 - 1/m^d$ by Claim 3.7, and so the algorithm is correct with at least this probability. ∎

We note that the running time of the algorithm (as opposed to its query complexity) involves checking (or preparing answers in advance to) all possible $\mathcal{M}_{Q,\epsilon}$ resulting from M_Q. Although this seems to be exponential in n, it can be reduced considerably. However, since a better and faster algorithm is presented in the following, we omit the details.

3.2 More preliminaries – pseudomatrices and a Ramsey-like lemma

Our aim in this section is to develop the machinery necessary for constructing a 1-sided error test. From the definition of such a test, in most cases a 1-sided error test can reject only if it finds a witness within the actual sampled points. Thus we will need to construct a sampling scheme that provably contains a witness with high probability if the input matrix M is ϵ-far from the property. Note that the analysis of the 2-sided error test that is presented in Section 3.1 does not provide such a guarantee.

In the above algorithm use was made of the following fact: Given the (k, d) 'matrix of blocks' which are all tagged X, one could choose any entry (with the right label) from each block to find the counter-example. Such 'loose formations' of entries are formalized below, with the definition of pseudogrids and pseudomatrices.

Definition 3.9 *A (k, d)-*pseudogrid *is a subset $(x_i)_{i\in[k]^d}$ of the grid $G(n, d)$, such that if $i, j \in [k]^d$ do not share a coordinate, then $x_i \le x_j$ if and only if $i \le j$ (in the ordering of $G(k, d)$).*

Alternatively, $(x_i)_{i\in[k]^d}$ is a pseudogrid if and only if there exist $\{r_{a,s} : 1 \le a \le d, 0 \le s \le k\}$ (one may assume $r_{a,0} = 0$ and $r_{a,k} = n$) such that for every a and i the a'th coordinate of x_i is at least r_{a,i_a} but less than r_{a,i_a+1}. Namely, there exists a partition of $G(n, d)$ into k^d blocks, that is defined by the above set of intervals, such that $(x_i)_{i\in[k]^d}$ consists of exactly one member from each block (see Figure 1 for an illustration).

A pseudomatrix *in an (n, d)-matrix A consists of a (k, d)-pseudogrid in $G(n, d)$ and its labeling according to A. Given a (k, d)-pseudomatrix in A defined by the pseudogrid $(x_i)_{i\in[k]^d}$, its* label matrix *is the (k, d)-matrix B (which is not necessarily in itself a submatrix of A) consisting of the entries of the given pseudomatrix in the corresponding locations, that is, the one satisfying $B(i) = A(x_i)$ for all $i \in [k]^d$.*

See Figure 1 for an example of a $(3, 2)$-pseudomatrix with its label 3×3 matrix.

Pseudomatrices will be used to show the existence of counter-examples within the queried values in 1-sided tests. Their main property is established in the following:

Fact 3.10 *Let A be a (k, d)-pseudomatrix of M with a label matrix W. Assume that W contains a k size labeled poset F such that no two entries share a coordinate. Then M also contains F.* ∎

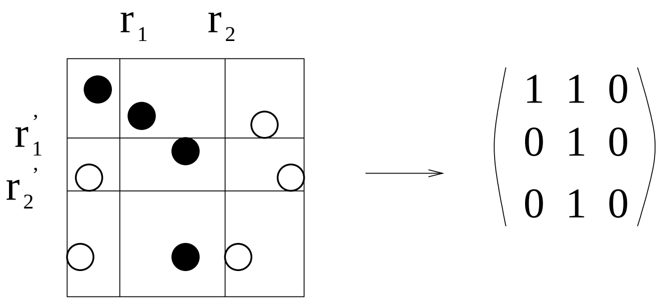

Figure 1: The 9 points in the grid on the left form a $(3,2)$-pseudogrid. Along with their labels (filled dots are 1's, hollow dots are 0's) they form a $(3,2)$-pseudomatrix. On the right is the label $(3,2)$-matrix of this pseudomatrix

For 1-sided algorithms, to guarantee the existence of a counter example, we will find a subset of the queries that will form a pseudomatrix whose restriction with each block of M is monochromatic. This is done by the following Ramsey-type lemma.

Lemma 3.11 *For every d and k there exists $c = c(k,d)$ such that for any $l \geq cm^c$, every 0/1-labeled (ml,d)-matrix which is partitioned into m^d blocks of size l^d contains an (mk,d)-pseudomatrix, so that its intersection with each of these blocks is a (k,d)-pseudomatrix whose entries are all identical.*

In order to prove the above lemma, we first need the following one, which deals with a 1-dimensional context. From now on we will mainly refer to the alternative definition of pseudogrids and pseudomatrices that involves the partition $\{r_{a,s} : 1 \leq a \leq d, 0 \leq s \leq k\}$. In the sequel the locations in strings of length l are numbered by $0, \ldots, l-1$.

Lemma 3.12 *For every k and every set of m binary strings of length $l \geq (2m+2)^{k/2} - 1$, there exist $0 = r_0 < r_1 < \cdots < r_k = l$ such that each string contains a monochromatic substring (i.e. a substring all of whose entries are of identical values) with exactly one entry in the interval $[r_i, \ldots, r_{i+1} - 1]$ for every $i = 0, \ldots, k-1$.*

Proof: We prove the claim first for odd values of k. The proof is by induction. The basis $k = 1$ is trivial (it holds even for m strings of length 1).

Let us denote $l_k = (2m+2)^{k/2} - 1$ and assume that the theorem was proven for $k-2$. We now prove it for k. We assume that none of the strings is monochromatic, as a monochromatic string will have a corresponding monochromatic substring for any $0 = r_0 < r_1 < \cdots < r_k = l$.

We first set the values of r_1 and r_{k-1} as follows: Let $l' = l_{k-2}$, choose r_1 in the interval $[1, \ldots, l-1-l']$ uniformly at random, and set $r_{k-1} = r_1 + l'$.

For every fixed string s let L_0 be the location of its first ('leftmost') '0' entry, L_1 the location of its first '1' entry, and R_0 and R_1 the locations of its last ('rightmost') '0' and '1' entries respectively. We are interested in the event that none of $\{L_0, L_1, R_0, R_1\}$ fall into the interval $I = [r_1, \ldots, r_{k-1} - 1]$.

The significance of this event is that if it occurs for a string s, then for any entry v in the interval $[r_1,\ldots,r_{k-1}-1]$ there are at least two entries of s with an identical label as v, one in the interval $[0,\ldots,r_1-1]$, and one in the interval $[r_{k-1},\ldots,l-1]$.

Since one of L_0 and L_1 is equal to 0 and one of R_0 and R_1 is equal to $l-1$, at most two of the four numbers above may fall into the interval I. Calculating the probability that any of L_0, L_1, R_0 and R_1 is in the interval I, it is clearly bounded by $2\frac{l'}{l-1-l'}$. But if $l \geq l_k$ then $\frac{l'}{l-1-l'} < \frac{1}{m}$. In particular, for such an l there exist r_1 and respectively r_{k-1} for which the above event does not happen for any of the m strings. This means that for every string from the set and any of its entries in the interval $[r_1,\ldots,r_{k-1}-1]$, there exist identical entries in the interval $[0=r_0,\ldots,r_1-1]$ and in the interval $[r_{k-1},\ldots,r_k-1=l-1]$.

We now consider for each string its substring of length $l'=l_{k-2}$ consisting of the entries in the interval $[r_1,\ldots,r_{k-1}-1]$. To these we apply the induction hypothesis, and find $r_2,\ldots,r_{k-2}$ so that each string contains a monochromatic substring with one entry in the interval $[r_i,\ldots,r_{i+1}-1]$ for every $1 \leq i < k-1$. By the above discussion, this substring can be extended to a monochromatic substring of the original string by finding identically labeled entries, one in the interval $[r_0,\ldots,r_1-1]$ and one in the interval $[r_{k-1},\ldots,r_k-1]$. This concludes the proof for odd k.

The proof for even values of k is identical to that for odd values except for the base case of $k=2$ which can easily be verified. ■

As an aside, before we continue we note that the bounds of Lemma 3.12 cannot be made linear in m.

Observation 3.13 *There is a set of $2m-2$ strings of length m^2, for which there exist no $0=r_0<r_1<r_2<r_3<r_4=m^2$ which satisfy the assertion of Lemma 3.12 for $k=3$.*

Proof: For every $1 \leq i \leq m-1$ let s_i denote the string consisting of im '1' entries followed by $(m-i)m$ '0's, and let t_i denote the string consisting of of the concatenation of m copies of the string with i '1's followed by $m-i$ '0's. Our set is the union of $\{s_i : 1 \leq i \leq m-1\}$ and $\{t_i : 1 \leq i \leq m-1\}$.

Now, assuming that $r_0 < \cdots < r_4$ exist, we divide into cases according to whether there exists j such that $jm \leq r_1 < r_3 \leq (j+1)m$. If there exist such j, then let $i = r_2 - jm$ and note that the string t_i has no substring satisfying the requirements, as it contains only '1's between r_1 and r_2, and only '0's between r_2 and r_3.

If there exists no j as above, then there exists i such that $r_1 \leq im \leq r_3$. In this case s_i has no substring satisfying the requirements, as it contains only '1's between r_0 and r_1, and only '0's between r_3 and r_4. ■

We now turn back to the proof of the main lemma.

Proof of Lemma 3.11: We use here the alternative definition of pseudomatrices, about the existence of a partition given by $\{r_{d',j} : 1 \leq d' \leq d, 0 \leq j \leq mk\}$ such that $(x_i)_{i\in[mk]^d}$ consists of one member from each block. In our case, the partition also has to satisfy $r_{d',k\cdot j} = l \cdot j$ for every

$0 \leq j \leq m$ to ensure that the intersection of the (mk, d)-pseudomatrix with each of the m^d blocks of A is a (k, d)-pseudomatrix.

The proof is by induction on d. We shall actually prove the (seemingly stronger) statement that given p such matrices (for any $p > 0$) with $l \geq c(mp)^c$, there exists a corresponding pseudomatrix for each of them, all with the same partition $\{r_{d',j} : 1 \leq d' \leq d, 0 \leq j \leq mk\}$ of $[n]^d$. The basis, $d = 1$, follows directly from Lemma 3.12, by cutting each string into m equal length substrings and applying Lemma 3.12 to these $p \cdot m$ new strings.

Given that the lemma is known for the $(d-1)$-th dimensional case, we now show it for the d dimensional case. Suppose that we are given p matrices $A_0, \ldots, A_{p-1}$ over $[ml]^d$ for which we want to find the partition. Note that every matrix can actually be viewed as being a collection of m different matrices by dissecting the matrix along the dth dimension into m parts, each of which is an $[ml]^{d-1} \times [l]$ matrix, see Figure 2. In turn, each of the m matrices resulting from this dissection can be thought of as an $[ml]^{d-1}$ matrix where each entry is a binary string of length l.

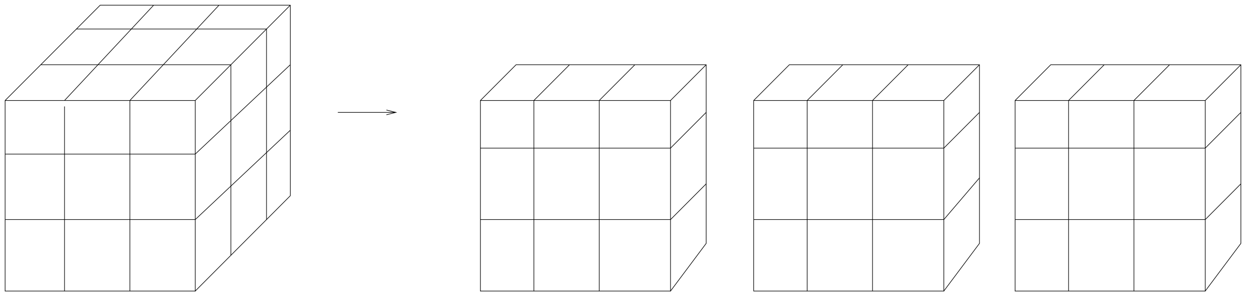

Figure 2: On the left a $(3l,\ 3)$-matrix with 3^3 blocks. It is viewed as 3 matrices, each with 3^2 blocks, by 'slicing' the matrix along the third dimension.

The basic idea of the proof is the following: We look at the individual $(ml)^{d-1}$ strings that are the entries to those mp matrices, and apply Lemma 3.12 to find a collection of $r_{d,j}$'s so that each string has a k-size monochromatic substring with exactly one point in each interval $[r_{d,j-1}, \ldots, r_{d,j} - 1]$. Once this is done, we may replace each l-length string with the single value that is identical to the value of the monochromatic substring that is found. This reduces each of the d-dimensional matrices to a $(d-1)$-dimensional matrix. Finding a common partition for these mp matrices will end the proof. By the induction hypothesis we obtain such a partition as we are now in the $(d-1)$-dimensional case.

The only technical problem in the above outline is that the number of strings we begin with is too large to apply Lemma 3.12, and hence we first pick a suitable sub collection. Formally this is done as follows: First, for every $0 \leq s < p$, $i = (i_1, \ldots, i_{d-1}) \in [ml]^{d-1}$ and $0 \leq q' < m$, we consider the string of length l consisting of the entries of A_s in locations $\{(i_1, \ldots, i_{d-1}, q'l + q) : 0 \leq q < l\}$. We refer to such string as the string indexed by $i = (i_1, \ldots, i_{d-1})$. This is a collection of $pl^{d-1}m^d$ strings of size l, corresponding to the entire collection of $(md)^{d-1}$ lines along the d'th dimension in each of the $p \cdot m$ matrices obtained after we have 'sliced' each of the p original matrices into m matrices. As this collection is of a size too large to apply Lemma 3.12 according to our plan, we just restrict ourselves to a subset of them: For some $l' < l$, we consider only strings that are indexed by $i = (i_1, \ldots, i_{d-1})$ for which every coordinate modulo m is between 0 and $l' - 1$. This will yield a collection of $pl'^{d-1}m^d$ strings. We do this for $l' = c(k,d-1)(pm{\cdot}m)^{c(k,d-1)}$, so in particular to ensure that $l' \le l$, we require that

$$c(k,d-1)(pm{\cdot}m)^{c(k,d-1)} \le l \tag{1}$$

For every fixed $0 \le q' < m$, we now want to find $r_0, \ldots, r_k$ that will partition all $m' = p(l'm)^{d-1}$ strings consisting of the entries of A_s in $\{(i_1, \ldots, i_{d-1}, q'l + q) : 0 \le q < l\}$ (for any $0 \le s < p$ and the values for $i_1, \ldots, i_{d-1}$ considered above) according to Lemma 3.12. As we do this for each of the m collections that is defined by different q's independently, this requires that:

$$l \ge (2m'+1)^{k/2} - 1 = (2p(l'm)^{d-1})^{k/2} - 1 \tag{2}$$

For each $0 \le s < p$, $0 \le q' < m$ and $i = (i_1, \ldots, i_{d-1})$ we take the common label of the substring found by Lemma 3.12 to be the entry of a new $(ml', d-1)$-matrix $A'_{ms+q'}$, at location $l'\lfloor i/l \rfloor + (i \bmod l)$ (we "delete" in the definition of the new matrices the strings which were not considered when finding $r_0, \ldots, r_k$). On these $(ml', d-1)$-matrices, $A'_0, \ldots, A'_{pm-1}$, we invoke the induction hypothesis to find the values $\{r'_{d',j} : 1 \le d' \le d-1, 0 \le j \le mk\}$, and the corresponding $(mk, d-1)$-pseudomatrices. Now set $r_{d,q'k+j} = q'l + r_j$ for $0 \le j \le k$ (remember that $r_0 = 0$ and $r_k = l$), which defines $\{r_{d,j} : \ 0 \le j \le mk\}$ of the pseudomatrix partition along the d'th dimension. These clearly refine the original partition of the matrices into blocks. We then set $r_{d',j} = l\lfloor r'_{d',j}/l' \rfloor + (r'_{d',j} \bmod l')$ for $1 \le d' \le d-1$, and note that the refinement of the block partition of $A'_0, \ldots, A'_{pm-1}$ translates to a refinement of the block partition of $A_0, \ldots, A_{pm-1}$. In order to find the required (mk, d)-pseudomatrix in A_s for any $0 \le s < p$, we can consider the pseudomatrices found in $A'_{ms}, \ldots, A'_{ms+m-1}$, and substitute each of their entries with the corresponding monochromatic string of length k found earlier in A_s that was used to set the entry.

The only thing that remains to be done is to ensure that the two conditions (1) and (2) above on the relations between l, m and l' are met, which is not hard for an appropriate choice of c in the requirement that $l \ge c(mp)^c$. ■

3.3 A 1-sided test

Here we present the 1-sided test for $\mathcal{M}_F$. We essentially follow the 2-sided error algorithm described in Subsection 3.1. The problem is that in the case that there are less than an $\epsilon/2$ fraction of the blocks that are tagged by X, the above algorithm may err in both ways as its correctness is based on the assumption that the event E (of correctly detecting the X-blocks) happens. To overcome this we will make additional queries in each block to assure us that if the approximated matrix is far from satisfying the property, then a counter-example to the property exists within the points already queried. In turn, this also avoids the necessity of checking all (n, d)-matrices approximable by the queried values, as we need only to check the queries themselves. This makes the running time of the algorithm polynomial in ϵ^{-1} too.

Theorem 3.14 *For every fixed $\forall$ poset property of (n,d)-matrices, with $d = O(1)$, there exists a 1-sided error ϵ-test which makes poly$(\frac{1}{\epsilon})$ many queries, independently of the size of the input.*

We shall use the Ramsey-like lemma presented in the previous section. In order for us to be able to work with it, we need to consider only embeddings in which no two entries share a coordinate, so pseudomatrices can be considered. The following lemma, in the vain of Lemma 3.5, grants this; the proof is also very similar to that of Lemma 3.5 and is therefore omitted.

Lemma 3.15 *Let the grids $G(mk, d)$ and $G(ml, d)$ be considered as structures over $G(m, d)$ in which every entry is a (k, d)-block and an (l, d)-block respectively. If a k-size poset P embeds into $G(ml, d)$, then there is an embedding of P into $G(mk, d)$ so that the two embeddings map each point of P to the same respective block in $G(ml, d)$ and $G(mk, d)$, the mappings are isomorphic on corresponding blocks (in terms of the poset relation), and moreover no two points share a coordinate in $G(mk, d)$.*

Proof of Theorem 3.14: We present the 1-sided error test. We start exactly as in the algorithm of Subsection 3.1 by dividing M into m^d blocks of equal size, for $m = (\frac{2^{k+1}k}{\epsilon^{k+1}})^{d-1}+1$. We query $\frac{8d\ln m}{\epsilon}$ uniformly random queries independently in every block of M and tag each block as being 1, 0 or X according to the queries made: '1'/ '0' if all values that were queried inside the block are 1/0 respectively, and X otherwise. If there are at least an $\epsilon/2$ fraction of the blocks that are X-tagged then we answer 'No'.

If there are less than an $\epsilon/2$ fraction of the blocks that are X-tagged then we further divide each block into $l^d = (c \cdot m^c)^d$ sub-blocks, where $c = c(k, d)$ is the constant of Lemma 3.11, and query one arbitrary query in each sub-block. We now set the outcome of the test as follows: We answer 'No' if there is a counter example among the queried points and answer 'Yes' otherwise (meaning that M is close to having the property).

Clearly the overall query complexity of this algorithm is polynomial in ϵ^{-1} (for a fixed d and k) and independent of n. We now prove its correctness.

Claim 3.16 *The algorithm is a 1-sided error algorithm with an error probability bounded by $1/m^d$.*

Proof: To show correctness, let us analyze the various cases in which the algorithm may end. If after the first round of queries there are at least an $(\epsilon/2)$-fraction of the blocks that are X-tagged, then, exactly as in the proof of the 2-sided test, a counter example to the property is guaranteed to exist already within the queried locations (with probability 1). Hence in this case, the algorithm answers 'No' and is correct with probability 1.

Assume then that there are less than an $\epsilon/2$ fraction of the blocks that are X-tagged. We then have the second phase of queries in each sub-block after which we re-tag each block according to the union of the old and new queries in it. Namely, some old 1-blocks and/or 0-blocks may become X-blocks. Again we may assume that there are less than an $\epsilon/2$ fraction of the blocks that are X, as otherwise we are back in the first case. Define now the (ml, d)-matrix M_Q, in which every entry corresponds to a sub-block of M that is labeled by 0/1 as determined by the value of the query in this sub-block. By Lemma 3.11 there is an (mk, d)-pseudomatrix W, containing a monochromatic (k, d)-pseudomatrix in each of the m^d blocks of M_Q. Note that in the natural correspondence between blocks of W and blocks of M, the label of every $0/1$ block of M is identical to the label of the corresponding block of W (on X-blocks of M the label of the corresponding block of W may be either '0' or '1'). We claim that even when we consider only the entries queried in the sub-blocks that appear in W, and decide to accept or reject according to whether they contain a counter example, this decision is correct with very high probability (and is 1-sided). Indeed, if a counter example is found among these points then certainly the algorithm rejects correctly.

We now claim that if the event E happened (as in Definition 3.6) and W does not contain a counter example, then M is ϵ-close to having the property. This will conclude the proof of the algorithm, because the event E occurs with probability at least $1 - 1/m^d$, and so the algorithm will be correct with at least this probability (as in addition to this the algorithm never incorrectly rejects the input).

Assume then that W does not contain a counter example and that E has happened. We show how to obtain a matrix M_W from M that has no counter example by changing at most ϵn^d entries of M. First we change the entries in every $0/1$ block of M to have the label of the corresponding block. As in Section 3.1, we may assume that this will incur a change in at most $\frac{\epsilon}{4}n^d$ of the entries. We then change every entry in an X-block to have the label of the corresponding intersection with W. As there are at most an $(\epsilon/2)$ fraction of X-blocks, this may result in at most $\frac{\epsilon}{2}n^d$ additional changed entries. Hence we get a matrix M_W, that is at most $\frac{3\epsilon}{4}$-far from M.

We now claim that M_W has no counter example. Indeed, assuming that M_W contains a counter example P, by Lemma 3.15 (looking at M_W as an (ml, d)-matrix with $l = n/m$) there is a counter example in an (mk, d)-matrix in which its m^d-size blocks are labeled as the blocks of M_W. Moreover, no two points of this counter example share a coordinate, and so its existence in a label matrix of some pseudomatrix implies its existence in the pseudomatrix itself. Now the label matrix of W is such an (mk, d)-matrix, and W in itself is a pseudomatrix contained in M_Q. Hence M_Q contains P with no two points sharing a coordinate. However, each entry of M_Q corresponds to a sub-block of the actual input M that contains at least one queried point that is labeled by the same label as the corresponding entry of M_Q. We therefore conclude that there is a counter example in the queried points of M in the sub-blocks that correspond to the points of W, contrary to our assumption. ■

An additional remark is due here: The algorithm was described as if it is adaptive, however as queries at the second stage do not depend on the answers at the first stage, the algorithm is in fact non-adaptive (we can clearly first make the queries of both stages, and only then count the number of the X-blocks to decide whether the matrix should be rejected on the grounds of having too many of them).

4 Testing of non-binary matrices

In this section we extend the result of the previous section to include forbidden poset properties for matrices that are not $0/1$, but have entries from a fixed finite alphabet Σ. Such properties are natural extensions of $0/1$-matrix properties.

The main result here is:

Theorem 4.1 *For every fixed $\forall$ poset property of (n, d)-matrices over a fixed finite alphabet and with $d = O(1)$, there exists a 1-sided ϵ-test whose number of queries and running time are doubly exponential in poly(ϵ^{-1}), and are independent of the size of the input.*

The proof uses some general ideas from the previous section, and in particular the idea of arriving to a partition into blocks for which most of them have almost no 'internal features' (similarly to them being nearly monochromatic in the 0/1 case). It turns out, however, that this situation is much harder then the 0/1 case. We will first need some additional definitions and machinery.

4.1 Homogeneity in partitions and a strengthening of Lemma 3.11

Given a partition of the input matrix into blocks, and a set of queries from each block, unlike in the 0/1 case, it cannot be guaranteed that if more than one type of label was found in many of the blocks then a counter-example to a given property $\mathcal{M}_F$ exists. It could be for example that a 0/1/2 matrix property is defined by a forbidden collection of labeled posets F, all of whose members contain the label '2', while the partition of the input matrix into blocks may contain many blocks with both 0's and 1's but no 2. Therefore, a notion of a block being monochromatic is replaced with a more general notion of being 'featureless'.

For the rest of this section Σ will always be a finite alphabet of size h. In the new framework, we consider two partitions of a matrix at a time. We consider as before a partition P of M into m^d blocks, and a refinement achieved by repartitioning each block of the first partition into l^d sub-blocks, thereby obtaining a partition Q of M into $(ml)^d$ blocks. We label every block and every sub-block with a subset of the alphabet Σ. In the construction of the tester this subset corresponds to the set of all labels found while querying entries from this block. Note that in particular a sub-block is always labeled with a subset of the set used to label the whole block.

We will treat a block of P as *featureless* if all its sub-blocks have the same label. However, in some cases it is only possible to find blocks that are close to satisfying this, which motivates the following definition.

Definition 4.2 *A labeled set is called* ϵ-homogeneous *if all but at most an ϵ fraction of its labels are identical. For such a set we call this common label the* $(1-\epsilon)$-dominant label *of the set.*

Given a partition P of the matrix M and a refinement Q of P, and given a labeling as above of these partitions, we call a block of P ϵ-homogeneous *if its label in P is the $(1-\epsilon)$-dominant label for the set of the corresponding sub-blocks in Q with their labels.*

Just as Lemma 3.11 was used in Section 3 to find a matrix which is both close to M and simple to check for $\mathcal{M}_F$, we shall use a similar lemma here. The pseudomatrix which we will find will be 'monochromatic' in each block, in the sense that all corresponding sub-blocks will have the same label (which is a subset of Σ). However, here we also need to ensure that for most blocks which are homogeneous enough, the sub-blocks will have the label of the block, so we will not need to modify many of the entries of M inside these blocks to arrive at the simplified matrix. We thus need the following strengthening of Lemma 3.11.

Lemma 4.3 *For every h, d and k there exists $c = c(h,k,d)$ such that for any $l \geq cm^c$, every (ml,d)-matrix W labeled with a set of h labels which is partitioned into m^d blocks of size l^d contains an (mk,d)-pseudomatrix satisfying the following. The intersection of the pseudomatrix with each of the blocks of W is a (k,d)-pseudomatrix whose entries are all identical, and moreover for all but at most an ϵ fraction of the $\frac{1}{c}\epsilon^c$-homogeneous blocks of W (for any $\epsilon > 0$), the common label of the intersection with them will be identical to the corresponding dominant label.*

The proof here follows closely the outline of the proof of Lemma 3.11, with additional care taken for the homogeneous blocks (the generalization to non-binary alphabets is not hard by itself). In the proof of this lemma, we also need a corresponding strengthening of Lemma 3.12, as formulated in the following definition.

Definition 4.4 *Let M be a collection of strings over Σ. For $\delta < 1$ we denote by $DOM_M(\delta)$ all strings in M that have a $(1-\delta)$-dominant label. For each such string $s \in DOM_M(\delta)$ we denote by dom_s the appropriate dominant label.*

Lemma 4.5 *For every integers h, k and $\epsilon < 1$ there exists $b = b(h,k,\epsilon)$ such that for any set M of m strings over Σ, of length $l \geq bm^b$, and every set $\mathcal{S} \subseteq DOM_M(1-\epsilon/b)$, there exist a sequence of indices $R = (0 = r_0 < r_1 < \cdots < r_k = l)$ such that:*

1. *Every string $s \in M$ contains a monochromatic substring (i.e. a substring all of whose entries are identical) with exactly one entry in the location range $r_i, \ldots, r_{i+1}-1$ for every $0 \leq i \leq l-1$. We denote this substring as $s(R)$ and its label by $lab(s(R))$.*
2. *For all but at most an ϵ fraction of the strings in $s \in \mathcal{S}$, $lab(s(R)) = dom_s$*

Proof: The proof here is by induction on k, namely we assume the existence of $b(h,k-2,\epsilon')$ for all ϵ' and show the existence of $b(h,k,\epsilon)$. This is done almost identically to the proof of Lemma 3.12, with extra care taken to ensure the second condition of the Lemma. The basis $k=1$ is also trivial here.

Assuming that the existence of $b(h,k-2,\epsilon')$ was shown, we show the existence of $b(h,k,\epsilon)$ (for any $\epsilon < 1$): Let $l' = b(h,k-2,\epsilon/2)m^{b(h,k-2,\epsilon/2)}$. Let $b > b(h,k-2,\epsilon/2)$ be an integer to be specified later (with the appropriate choice, this will be the suitable $b(h,k,\epsilon)$), let M be a collection of strings over Σ of length $l \geq bm^b > l'$, and let $\mathcal{S} \subseteq DOM_M(1-\epsilon/b)$. We will show that for an appropriate choice of b there is a sequence of indices as required.

Similarly to the proof of Lemma 3.12, let r_1 be randomly chosen in the range $1, \ldots, l-1-l'$ and set $r_{k-1} = r_1 + l'$. For every string $s \in M$ and every $x \in \Sigma$ we set $L_x(s)$ and $R_x(s)$ as the locations of its first occurrence and its last occurrence in s. If x does not appear in s we set $L_x(s) = -1, R_x(s) = n+1$.

For a string $s \in M$ and a given choice of r_1 let $A(s)$ be the event that none of $L_x(s), R_x(s)$ falls into $r_1, \ldots, r_1 + l' - 1$ for every $x \in \Sigma$. For every $s \in \mathcal{S}$, we denote by $B(s)$ the event that $s' = s[r_1], \ldots, s[r_1 + l' - 1]$ has dom_s as its $(1-\epsilon/(2b(h,k-2,\epsilon/2)))$-dominant label, where $s[i]$

denotes the ith place in s. Finally, let A be the event $\bigcap_{s \in M} A(s)$ namely, the event that $A(s)$ happens for every s.

Our aim is to show that: (1) $Prob[A]$ is large enough. (2) for every $s \in \mathcal{S}$, $Prob[B(s) \mid A]$ is large enough. Then, this would imply that there is a choice of r_1 for which the following two conditions hold simultaneously: (1') A is true and (2') for most $s \in \mathcal{S}$, $B(s)$ is true. This will imply that if we apply the induction hypothesis with $k-2$ on the substrings defined by the range $r_1, \ldots, r_1 + l' - 1$, we get $(k-2)$-length monochromatic substrings (for each string) that can be augmented into k-length monochromatic substrings in all strings, by (1'). Also, for the suitable fraction of strings that have $(1 - \epsilon/b(h, k-2, \epsilon/2))$-dominant labels, the substring found has the required dominant label. But, by (2') those strings come from (suitably many) original strings that have the same consistent $(1 - \epsilon/b)$ dominant label. This proves the additional requirement on the label of the monochromatic strings that are found. The details follow.

Let $s \in M$. As there are $2h$ events of the form $\{r_1 \le L_x < r_1 + l', r_1 \le R_x < r_1 + l'\}$, $x \in \Sigma$, it follows that $Prob[A(s)] \ge 1 - \frac{2hl'}{l-l'-1}$. We will require that l is large enough to satisfy

$$\frac{2hl'}{l - l' - 1} \le \epsilon/8m \tag{3}$$

which will imply that $Prob[A] \ge 1 - \epsilon/8$.

Let $s \in \mathcal{S}$, and let $\alpha(r_1)$ be the number of occurrences of dom_s in $s[r_1], \ldots, s[r_1 + l' - 1]$. Then clearly $\Sigma_{r_1=1}^{l-l'-1} \alpha(r_1) \ge ((1 - \frac{\epsilon}{b})l - 2l')l'$, hence the expected value of $\alpha(r_1)$ satisfies:

$$E[\alpha(r_1)] \ge \frac{(1 - \epsilon/b)l - 2l'}{l - l' - 1} \cdot l' \ge (1 - \frac{\epsilon}{b} - \frac{2l'}{l - l'})l'$$

.

By Markov Inequality this implies that $Prob[B(s)] \ge 1 - \beta$ where $\beta = (\frac{\epsilon}{b} + \frac{2l'}{l-l'})/(\epsilon/(2b(h, k - 2, \epsilon/2))$. We require that l' be such that β satisfies:

$$\beta \le \epsilon/8 \tag{4}$$

Conditions (3) and (4) imply that for every $s \in \mathcal{S}$, $Prob[A \cap B(s)] \ge 1 - \epsilon/4$. Hence this implies that there is a choice of r_1 for which A is true, and $B(s)$ is true for at least a $(1 - \epsilon/4)$-fraction of the $s \in \mathcal{S}$. Let us fix such r_1 and denote this collection of strings for which $B(s)$ holds by $\mathcal{S}_1$.

Let $\epsilon' = \epsilon/2$, let M' be the collection of strings obtained from M by $M' = \{s[r_1], \ldots, s[r_1 + l' - 1] \ : \ s \in M\}$. Let $\mathcal{S}'$ be those strings in M' that originated from the corresponding strings in $\mathcal{S}_1$. By the definition of $\mathcal{S}_1$ it follows that $\mathcal{S}' \subseteq DOM_{M'}(1 - \epsilon'/b(h, k-2, \epsilon'))$. Hence, by the induction hypothesis on M' and $\mathcal{S}'$ for $k-2$ and ϵ', we get that there are indices $r_2 \le \cdots \le r_{k-2}$ for which conditions (1) and (2) of the Lemma hold. We set $R = r_1, r_2, \ldots, r_{k-2}, r_1 + l' - 1$, namely, we append r_1 and $r_1 + l' - 1$ as the initial and ending indices to the sequence obtained by induction for $k-2$. Then, as A holds for r_1 then certainly condition (1) of Lemma holds for R and M. Also, as condition (2) holds for $\mathcal{S}'$ it follows that for at least a $(1 - \epsilon/2)$-fraction of the strings in $\mathcal{S}_1$, the monochromatic substring $s(R)$ has label identical to $dom(1 - \epsilon/b)$. However as $\mathcal{S}_1$ is of size at least

$(1 - \epsilon/2)$ of the size of $\mathcal{S}$ it follows that the number of strings in $\mathcal{S}$ for which condition (2) of the lemma holds (for k) is at least $(1 - \epsilon/2)^2 \geq 1 - \epsilon$.

Finally, it remains to be checked that there is a constant $b = b(h, k, \epsilon)$ such that if $l \geq b \cdot m^b$ then l and l' satisfy conditions (3) and (4), which is easily verified from their formulation. This concludes the proof for odd values of k. For even k the proof directly follows form the case of $k + 1$. ■

Proof of Lemma 4.3: This proof is similar to the proof of Lemma 3.11, and uses induction on d. Similarly to Lemma 3.11, we actually prove the existence of a common partition for a set of p such matrices $A_0, \ldots, A_{p-1}$, given that $l \geq c(pm)^c$, so that the partition has the corresponding pseudomatrix in each matrix, with the additional property that the fraction of blocks with a dominant label whose intersection with the pseudomatrices has a different label is less than ϵ (summing over all p matrices; this is not necessarily true for each matrix separately). The basis $d = 1$ follows easily from Lemma 4.5.

Assume that the lemma is proven for $d-1$ and that $c(h, k, d-1)$ is known. Let $\delta = \frac{1}{c}\epsilon^c$, where $c = c(h, k, d)$ will be defined later. As in the proof of Lemma 3.11, for every A_s, for $0 \leq s < p$, we consider only part of the $pm \cdot (ml)^{d-1}$ strings of size l that are defined by 'slicing' every A_s, along the d'th dimension, into m matrices of size $l \cdot (ml)^{d-1}$ (see again Figure 2). For Lemma 3.11, this partial set was arbitrary picked by looking at some fixed residues of the coordinates. Here our goal is to find a (not too large) subset of the strings so that for all but a $\delta^{1/2}$ fraction of them the following will hold: If a string comes from a block which contains a δ-dominant label, then this label is also a $\delta^{1/2}$-dominant for the string. To construct such a set of strings of size $pm(ml')^{d-1}$, let $L_i \in [l]$ be a random set of size l' for each $1 \leq i \leq d$. We consider only the lines for which for every $1 \leq i \leq d-1$, its i-th coordinate is in L_i. With positive probability the fraction of such strings which come from blocks which contain a δ-dominant label, but for which the same label is not $\delta^{1/2}$-dominant, is less than $\delta^{1/2}$. We thus fix $L_1, \ldots, L_d$ for which this holds.

We now want to apply Lemma 4.5, to get a partition of $[l]$ into k parts so that there is a monochromatic substring of size k in each of the strings, with exactly one entry in each part. To apply Lemma 4.5 we need $l \geq b(pm(ml')^{d-1})^b$, where b is the constant from Lemma 4.5. This partition of $[l]$ defines the partition for the d'th coordinate. We now construct matrices $A'_0, \ldots, A'_{pm-1}$, using the labels of the monochromatic substrings found, exactly as in the proof of Lemma 3.11. The fraction of the strings not conforming to a dominant label of their block is bounded by $b\delta^{1/2}$. Hence, the fraction of the blocks which correspond to blocks with δ-dominant labels in the original matrix, but do not have the same labels as $\delta^{1/4}$-dominant ones, is bounded by $b\delta^{1/4}$.

On this set of pm matrices we invoke the induction hypothesis to find partitions along the other $d-1$ dimensions, $\{r'_{d',i} : 1 \leq d' \leq d-1, 0 \leq i \leq mk\}$, exactly as in the proof of Lemma 3.11. For this we choose $l' = c(h, k, d-1)(pm \cdot m)^{c(h,k,d-1)}$.

Set $c_1 = 2c(h, k, d-1)$. We note that up to an $b\delta^{1/4} + c_1^{(1/c_1)}\delta^{1/4c_1}$ fraction of the blocks that had a δ-dominant label in $A_0, \ldots, A_p$ will not have the same label for the restriction of the appropriate pseudomatrix. Thus, there exists a choice of a large enough $c(h, k, d)$ that ensures that this is less than ϵ (remember that $\delta = \frac{1}{c}\epsilon^c$). This together with the previous two restrictions on l', l and m

sets an appropriate choice of $c = c(h, k, d)$. ■

4.2 The test

Proof of Theorem 4.1: Let F be a collection of forbidden Σ-labeled posets (for some fixed finite alphabet Σ), each of size at most k. In order to construct a 1-sided error ϵ-test for the property $\mathcal{M}_F$ we want to arrive at a situation similar to the 1-sided test for 0/1 properties. Namely, we want to have two partitions of the matrix into blocks P, Q, where Q is a refinement of P, such that most blocks of P are δ-homogeneous with respect to the sub-blocks as defined by Q (for some small constant δ). In this case we will be able to perform the test in a similar way to the 0/1 test. However, in order to arrive at such a situation, we cannot rely anymore on the assertion that if there are many non-homogeneous blocks then we have a counter example (as in the 0/1 case). Instead, we construct a sequence of partitions $P_0, \ldots, P_q$ of M, where each P_i is a refinement of the previous one. We will prove in what follows that for such a sequence, there are two consecutive members for which the above holds. The proof is by an iterative argument reminiscent of the proof of Szemerédi's Regularity Lemma ([24], see [7, Chapter 7] for a good exposition). However, the dependency of the number of queries on ϵ will not be as severe as might be expected from such an argument; for every fixed property it will be doubly exponential in a polynomial in ϵ, rather than a tower.

Formally, we choose $m_0 = 1$, and let $m_i = c \cdot (m_{i-1})^{c+1}$, where $c = c(h, k, d)$ is the constant provided by Lemma 4.3 and h is set to $2^{|\Sigma|}$. Let P_0 have just one block and let $P_1, \ldots, P_q$ be a sequence of partitions, each P_i being a partition of M into $(m_i)^d$ blocks which is a refinement of the previous one. We choose $q = |\Sigma| c(h, k, d)/(\frac{1}{3}\epsilon)^{1+c(h,k,d)}$, so in particular $m_q = exp(exp(poly(1/\epsilon)))$ for fixed Σ, d and k.

The algorithm proceeds as follows. First, $\frac{6}{\epsilon} \ln((m_q)^d \cdot |\Sigma|)$ uniformly random queries are made independently in each block of P_q. If a counter example is found among any of the query points the algorithm rejects, and otherwise it accepts.

It is clear from the formulation that the algorithm has a 1-sided error. We show in the sequel that it has bounded error for all inputs.

We define as E the event that no block of P_q (and hence no block of any $P_j,\ j < q$) contains more than a $\frac{1}{3}\epsilon$ fraction of entries of any label which did not appear in its queries. We note that with high probability E happens, and now prove, given that E happens, that if there is no counter example within the set of the queried points, then M is ϵ-close to having the property. To see this, we label each block of P_q with the set of all labels known to appear in it as a result of the queries made. We also label all the blocks of the other partitions (remember that P_q is a refinement of all of them) with the set of labels found while making queries within them. The next step is to choose some $0 \le p < q$, using the following claim.

Claim 4.6 *There exists $0 \le p < q$ for which all but at most a $\frac{1}{3}\epsilon$ fraction of the blocks of P_p are $\frac{1}{c(h,k,d)}(\frac{1}{3}\epsilon)^{c(h,k,d)}$-homogeneous with respect to P_{p+1}.*

Proof: For every $\sigma \in \Sigma$, $0 \leq l < q$, and a set of blocks A of P_l, we denote by $\chi_{\sigma,l}(A)$ the fraction of blocks of P_l in A whose label contains the color σ. Let $\chi_l(A) = \Sigma_\sigma \chi_{\sigma,l}(A)$ and let $\psi_l = \chi_l(P_l)$. By a slight abuse of notation we let $\chi_{\sigma,l+1}(A)$ and $\chi_{l+1}(A)$ denote the respective quantities where the set of the sub-blocks of A from P_{l+1} is used instead of A itself. We claim that for any $\delta, \gamma > 0$, if P_l has more than a δ-fraction of blocks which are not γ-homogeneous, then $\psi_{l+1} \leq \psi_l - \delta \cdot \gamma$.

Indeed, we observe that for any $p' \leq l$ and any set of blocks A of $P_{p'}$, we have $\chi_{p'}(A) \geq \chi_l(A)$. We also observe that $\psi_l = Prob[A]\chi_l(A) + (1 - Prob[A])\chi_l(P_l - A)$ (where $Prob[A]$ is the relative size of A to the whole matrix).

Now assume that P_l has more than a δ-fraction of blocks that are not γ-homogeneous, and let A be the set of all blocks of P_l that are not γ-homogeneous. Then $Prob[A] \geq \delta$. Also, by the above observation it is enough to show that $Prob[A](\chi_l(A) - \chi_{l+1}(A)) \geq \delta \cdot \gamma$. By definition, for any $p \geq l$, $\chi_p(A) = \sum_B Prob[B]\chi_p(B)$, where B ranges over the blocks in A and $Prob[B]$ denotes the relative size of B. However, for each block B, assuming that its label in P_l is S, $|S| = r \leq h$, we have that $\chi_l(B) = r$, while as B is not γ-homogeneous we have $\chi_{l+1}(B) \leq Prob[S] \cdot r + (1 - Prob[S]) \cdot (r-1) \leq r - 1 + Prob[S] \leq r - \gamma$ (here $Prob[S]$ is the fraction of blocks of P_{l+1} in B that are labeled S). Hence $Prob[A](\chi_l(A) - \chi_{l+1}(A)) \geq \delta \cdot \gamma$ as claimed.

To end the proof of the claim, note that since $\psi_0 \leq |\Sigma|$, if the claim does not hold for every p, it would imply that $\psi_q < 1$, which is a contradiction (as every block label must contain at least one member of Σ). Hence there exists p as in the formulation of the claim. ∎

We now return to the proof of the theorem: Choosing p as in the formulation of the claim, we proceed as follows. We consider the (m_{p+1}, d)-matrix consisting of the labels of the blocks of P_{p+1}, and its partition into $(m_p)^d$ blocks corresponding to the blocks of P_p. Over these we apply Lemma 4.3 to find a (km_p, d)-pseudomatrix so that its intersection with every block from P_p is a monochromatic (k, d)-pseudomatrix. By Lemma 4.3 for all but at most a $\frac{1}{3}\epsilon$ fraction of the homogeneous blocks in P_p, the restrictions of the pseudomatrix found to these blocks will have labels identical to the blocks. We let W be the label matrix corresponding to the above pseudomatrix.

We now consider the following (n, d)-matrix M_W: For every entry of M whose label is in the set which is the common label of the intersection of the above pseudomatrix with the block from P_p, this will also be the entry of M_W in the respective location. Every other entry shall be replaced with an arbitrary member of the set labeling the intersection of the pseudomatrix with the entry's block.

We note that, as in the 0/1 case, if E happens (as we assumed) then M_W differs from M in less than ϵn^d places, because the differences can only be in entries which did not conform to the corresponding block label in P_{p+1}, or entries which were in non-homogeneous blocks of P_p, or entries which were in homogeneous blocks of P_p whose label was not the common label of the intersection of the pseudomatrix found by Lemma 4.3 within this block. Since each of these three categories contains at most $\frac{1}{3}\epsilon n^d$ of the entries, M is ϵ-close to M_W.

We claim now that if the algorithm accepts then M_W has the property and hence we are done. Indeed, assume on the contrary that M_W contains a counter example. We claim that in this case there already exists a counter-example within the locations of the matrix queried by the algorithm (regardless of E): Given the existence of a counter example F_0 in M_W, by Lemma 3.15 there exists an embedding of the poset F_0 (disregarding the labels for now as they will be dealt with later) in M_W when considered as $G(m_p \cdot k, d)$, where each point in this embedding is in a different sub-block, and with no two points sharing a coordinate. As W is an $(m_p \cdot k, d)$-matrix, we look at this embedding in W. By our construction, the label of each entry in a P_p block of M_W is a member of the common block label of this block in M_W. Hence, in the embedding of F_0 in W, each entry is labeled by a set that contains the label of the original entry of F_0. However, as W is a label matrix of a pseudomatrix of P_{p+1} (where each block is considered as a point), an entry of M can be chosen from each block of P_{p+1} whose label is any desired member of the corresponding set-label of the point in W. In particular, we can choose for any entry the label that leads to an embedding of F_0 in M. ■

5 A $\forall\exists$ property that is not testable

In this section we construct a $\forall\exists\phi(x_1, \dots, x_k)$ property that is not ϵ-testable for some fixed ϵ. The construction is similar in spirit to the non-testable graph property constructed in [1]. We first construct such a property for a model which is stronger than the poset model, and then based on it derive a poset-only $\forall\exists$ property which is not testable. For the rest of this subsection we consider only 2-dimensional matrices, and use the notion of rows and columns in the usual matrix sense.

5.1 A non-testable property concerning submatrices

The property that we present here uses the alphabet $\{0, 1, 2\}$ and is in a slightly more general model than $\forall\exists$-poset. Let S_n be the symmetric group on n elements, let A be an $n \times n$ matrix. A matrix B is said to be a row/column permutation of A if it can be obtained from A by first permuting the columns of A by an arbitrary permutation $\pi \in S_n$ and then permuting the rows of the resulting matrix by an arbitrary permutation $\rho \in S_n$. We say that a $\{0, 1, 2\}$-matrix satisfies the property 'permutation' if it is a row/column permutation of a symmetric matrix with all 2's on its primary diagonal, and no 2's anywhere else. It is not hard to see that this is equivalent to the matrix satisfying the following three conditions:

1. For every matrix entry which is not '2' there is an entry on the same row and an entry on the same column which are both '2'.
2. For every '2' entry there is no other entry on the same row or column which is also '2'.
3. The matrix contains none of the following 2×2 matrices as a submatrix (to ensure that the original matrix was symmetric).

$$\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}; \quad \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}; \quad \begin{pmatrix} 0 & 2 \\ 2 & 1 \end{pmatrix}; \quad \begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix}$$

Using the above equivalent formulation, the property 'permutation' can be expressed as a $\forall\exists\phi$ type property where ϕ uses, apart from the order relation and the value relations ("the entry at location x is 0/1/2"), the two additional relations: "x_1, x_2 are on the same row" and "x_1, x_2 are on the same column" (this model has some interest in its own right, and is discussed further in the concluding comments as well as in [2]). We claim that property 'permutation' is not testable:

Proposition 5.1 *Property 'permutation' is not $\frac{1}{20}$-testable even by a two sided error adaptive algorithm making $o(n^{1/2})$ queries.*

Proof: Using Yao's principle [25], we define a probability distribution on inputs and show that any fixed deterministic algorithm that queries $d = o(n^{1/2})$ queries has an average error (according to the distribution on inputs) of more than $\frac{1}{3}$.

Let us first define a distribution P on positive inputs, a distribution N on negative inputs, and then the distribution D will be to choose with probability $\frac{1}{2}$ a member according to P and with probability $\frac{1}{2}$ a member according to N. P is defined by first choosing randomly and uniformly a symmetric matrix B with 2's on the primary diagonal and 0's and 1's everywhere else. The input matrix A is constructed from B by permuting its rows according to a permutation which is chosen uniformly at random.

N is defined by just letting A be a uniformly random 0/1 matrix. The support of N includes also inputs which are close to satisfying 'permutation', but a matrix selected according to N will almost surely be $\frac{1}{20}$-far from satisfying it as claimed below

Claim 5.2 $Prob_N[dist(A, \text{`}permutation') \leq \frac{1}{20} \cdot n^2] \leq exp(-n)$

Proof: The distribution N is uniform on all 2^{n^2} possible 0/1 matrices. Let A be a matrix that belongs to 'permutation'. Then, there are at most $C = \binom{n^2}{\epsilon n^2} \cdot 2^{\epsilon n^2}$ 0/1-matrices that are ϵ-close to A (that bound is obtained by choosing ϵn^2 places to change, and 0/1 values to change those places to). The number of matrices in 'permutation' is at most $2^{n(n-1)/2} \cdot (n!)^2$; this bound is obtained by choosing a symmetric 0/1 matrix with 2 on its diagonal, and then permuting it by an arbitrary pair of permutations (in fact one can easily show that this number is exactly $2^{n(n-1)/2} \cdot n!$, and that the distribution P picks a uniformly random matrix with this property). Hence the number of 0/1 matrices that are ϵ-close to 'permutation' is at most $D \leq C \cdot 2^{n(n-1)/2} \cdot (n!)^2$. Fixing $\epsilon = 1/20$ and using the standard approximation for C, we get that $C \leq 2^{0.35 \cdot n^2}$. Thus $D \leq 2^{n^2(0.5+0.35+o(1))}$ implying that the probability above is at most $2^{-(0.15+o(1)) \cdot n^2}$ ■

Now, let $\mathcal{A}$ be an adaptive deterministic algorithm for testing the above property, that queries $d = o(n^{1/2})$ many queries. Such an algorithm can be represented by a decision tree of height d, where each node of the tree represent a query and the leaves represent accept or reject decisions. For such trees, the following was proved implicitly in [15] and other works, and has an explicit proof in [12].

Lemma 5.3 *Suppose that there exists two distributions P and N on inputs, so that for any subset $Y = \{y_1, \ldots, y_d\}$ of size d of the domain and any $g : Y \to \{0, 1\}$, we have $\Pr_{N|_Y}(g) \leq (1 + o(1))\Pr_{P|_Y}(g)$. Then it is not possible for a decision tree $\mathcal{A}$ to distinguish with a bounded error probability between an input chosen according to P and an input chosen according to N.*

The formulation in [12] actually has a stronger condition, $(1-o(1))\Pr_{P|_Y}(g) < \Pr_{N|_Y}(g) < (1+o(1))\Pr_{P|_Y}(g)$, but it is not hard to see that the same proof works also for the condition above. In here we will use P from above, and $\tilde{N}$, which is the conditioning of N on the event that the chosen matrix is indeed $\frac{1}{20}$-far from satisfying 'permutation'. However, since this event in our case occurs with probability $1-o(2^{-d})$, it is clearly sufficient to prove the above condition for N instead of $\tilde{N}$.

We need the following easy claim.

Claim 5.4 *For a fixed set $I \subset \{1,\ldots,n\}$ of size k, let σ be a uniformly chosen random permutation over $\{1,\ldots,n\}$. Then, with probability at least $1-\binom{k}{2}\cdot\frac{1}{n}$ there are no $i,j \in I$ for which $\sigma(i)=j$.* ■

For a matrix chosen according to to N, the restriction of the input to any set Y of size d is the uniform distribution over sequences of d bits, and so for any $g : Y \to \{0,1\}$ we have $\Pr_{N|_Y}(g) = 2^{-d}$ (and hence $\Pr_{\tilde{N}|_Y}(g) \le (1+o(1))2^{-d}$). Now we analyze the restriction to Y of a matrix chosen according to P. Let I be the set of all indexes of the matrix entries found in Y (in the first or second coordinate of any entry). It is clear that $|I| \le 2d$, and so the event of Claim 5.4 occurs with probability $1-o(1)$. However, conditioned on this event, the restriction of the matrix to Y will again be a uniformly random distribution over sequences of d bits, as Y will contain no '2' entry and no two members of Y will be correlated by the matrix symmetry. Thus $\Pr_{P|_Y}(g) \ge (1-o(1))2^{-d}$, and together with the probability bound for $\tilde{N}$ this implies the conditions for Lemma 5.3 that yield the testing bound.

5.2 Submatrices, tight submatrices and witnesses

In order to construct a non-testable property of 0/1 matrices that is strictly in the $\forall\exists$-poset model we need some machinery that will be developed here.

Relations like "x and y are not on the same row or column" can be expressed in the poset model using additional variables and quantifiers. For example, it can be seen that if $x \le y$ (in the product ordering of the 2-dimensional matrix), then they do not share a row or a column if and only if there exist w and z which are incomparable and moreover satisfy $x \le w \le y$ and $x \le z \le y$. We call such a pair variables a *witness* for x and y not sharing a coordinate.

We can extend this further: The locations $(x_{i,j})_{i,j\in[k]}$ represent a subgrid (and their labels a submatrix) if and only if they have between them the order relation that a subgrid has, and furthermore there are no witnesses for any $x_{i,j}$ and $x_{i,j'}$ not sharing a coordinate, as well as for any $x_{i,j}$ and $x_{i',j}$ not sharing a coordinate. Note however that we cannot distinguish this way between a submatrix and its transpose.

We can also use witnesses to express other notions: If two points $x \le y$ are not equal but do share a coordinate, then x and y reside on consecutive values of the other coordinate if and only if there exist no z different from x and y for which $x \le z \le y$. This allows us to express the following definition within a first order poset property. $(x_{i,j})_{i,j\in[k]}$ are said to represent a *tight subgrid* if there exist i_0 and j_0 so that $x_{i,j} = (i_0 + i, j_0 + j)$ for every i and j. Similarly we define the notion of a *tight submatrix* using their labels. We note now that $(x_{i,j})_{i,j\in[k]}$ represent a tight subgrid (or its transpose) if and only if they satisfy the appropriate order relations, and furthermore there exist no witnesses showing that $(x_{i,j})_{i,j\in[k]}$ is not a subgrid and no witnesses showing that $x_{i,j}$ and $x_{i,j+1}$ or $x_{i,j}$ and $x_{i+1,j}$ (for any i and j) do not reside consecutively on the non-shared coordinate.

We are now ready to define the non-testable poset property.

5.3 A non-testable $\forall\exists$-poset property

In our definition we shall use the following matrices, which we call the three *guide matrices*. The idea would be to encode the $\{0, 1, 2\}$ labeling in the property 'permutation' defined in Subsection 5.1 by copies of these matrices.

$$G_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The property 'permutation-p' that is defined below describes matrices that have the property 'permutation', where each label $i \in \{0, 1, 2\}$ is replaced with the guide matrix G_i. Formally, it is defined as the input matrix satisfying the following conditions (actually each statement should be replaced by a counterpart which is invariant with respect to taking a transpose of the input matrix, but this has no essential effect on the following analysis as the guide matrices are not symmetric).

1. For every $(x_{i,j})_{i\in[5],j\in[10]}$ there either exists a witness showing that they do not form a tight subgrid, or the labels of the input matrix M in these locations are such that $(M(x_{i,j}))_{i\in[5],j\in[5]}$ is a guide matrix if and only if $(M(x_{i,j}))_{i\in[5],j\in[10]-[5]}$ is a guide matrix.

2. Similarly to the above, for every $(x_{i,j})_{i\in[10],j\in[5]}$ that form a tight subgrid, $(M(x_{i,j}))_{i\in[5],j\in[5]}$ is a guide matrix if and only if $(M(x_{i,j}))_{i\in[10]-[5],j\in[5]}$ is.

3. If $(x_{i,j})_{i\in[5],j\in[5]}$ form a tight subgrid and their labels form a guide matrix other than G_2, then the input matrix M contains a "1" on the same row as $x_{1,1}$ and a "1" on the same column as $x_{1,1}$ (this can be formulated in terms of the order relations of the locations of the 1's with respect to $x_{1,0}$, $x_{0,1}$, $x_{2,1}$ and $x_{1,2}$).

4. If $(x_{i,j})_{i\in[5],j\in[10]}$ form a subgrid, and moreover $(x_{i,j})_{i\in[5],j\in[5]}$ and $(x_{i,j})_{i\in[5],j\in[10]-[5]}$ are both tight, then $(M(x_{i,j}))_{i\in[5],j\in[5]}$ and $(M(x_{i,j}))_{i\in[5],j\in[10]-[5]}$ cannot be both G_2.

5. Similarly if $(x_{i,j})_{i\in[10],j\in[5]}$ form a subgrid, and $(x_{i,j})_{i\in[5],j\in[5]}$ and $(x_{i,j})_{i\in[10]-[5],j\in[5]}$ are tight, then $(M(x_{i,j}))_{i\in[5],j\in[5]}$ and $(M(x_{i,j}))_{i\in[10]-[5],j\in[5]}$ cannot be both G_2.

6. If $(x_{i,j})_{i\in[10],j\in[10]}$ form a subgrid for which $(x_{i,j})_{i\in[5],j\in[5]}$, $(x_{i,j})_{i\in[5],j\in[10]-[5]}$, $(x_{i,j})_{i\in[10]-[5],j\in[5]}$ and $(x_{i,j})_{i\in[10]-[5],j\in[10]-[5]}$ are tight, then $(M(x_{i,j}))_{i\in[10],j\in[10]}$ is not any of the following formations.
$$\left(\begin{array}{c|c} G_2 & G_0 \\ \hline G_1 & G_2 \end{array}\right); \quad \left(\begin{array}{c|c} G_2 & G_1 \\ \hline G_0 & G_2 \end{array}\right); \quad \left(\begin{array}{c|c} G_0 & G_2 \\ \hline G_2 & G_1 \end{array}\right); \quad \left(\begin{array}{c|c} G_1 & G_2 \\ \hline G_2 & G_0 \end{array}\right)$$

We first establish the connection between the properties 'permutation' and 'permutation-p'.

Given an $n \times n$ matrix A labeled by $\{0,1,2\}$, we say that a $5n \times 5n$ 0/1 matrix M is the *tiling* of A, if for every $0 \le i, j \le n-1$ the tight submatrix $(M(5i+i', 5j+j'))_{0\le i',j'\le 4}$ is equal to $G_{A(i,j)}$. In other words, M is formed by replacing each entry of A with the appropriate 5×5 guide matrix.

Claim 5.5 *If M is a tiling of a matrix A which satisfies 'permutation', then M satisfies the property 'permutation-p'.*

Proof: It is not hard to see that a tiling of any 0/1/2 matrix does not contain any tight submatrices which are equal to a guide matrix (or its transpose) apart from those of the form $(M(5i+i', 5j+j'))_{0\le i',j'\le 4}$. Thus, the first two conditions in the definition of 'permutation-p' are satisfied in M. The third condition now follows from the first condition in the definition of 'permutation' (note that G_2 is the only guide matrix with a '1' entry on its second row, as well as the only one to have a '1' on its second column). The fourth and fifth conditions in 'permutation-p' follow from the second condition in 'permutation', and the sixth condition in 'permutation-p' follows from the third condition in 'permutation'. This completes the claim. ■

Claim 5.6 *If M is a $5n \times 5n$ matrix which satisfies 'permutation-p' and has any tight submatrix of the form $(M(5i+i', 5j+j'))_{0\le i',j'\le 4}$ which is equal to a guide matrix, then M is a tiling of some $n \times n$ matrix A which satisfies 'permutation'.*

Proof: The first two conditions in the definition of 'permutation-p' imply that if M satisfies it and has any tight submatrix of the form $(M(5i+i', 5j+j'))_{0\le i',j'\le 4}$ equal to a guide matrix, then M is a tiling of some $n \times n$ matrix A (these conditions say in essence that a tight submatrix adjacent to a tight guide submatrix is a guide matrix itself). The last four conditions in 'permutation-p' guarantee in turn that A satisfies the three conditions in the definition of 'permutation'. ■

Theorem 5.7 *Property 'permutation-p' is not $\frac{1}{500}$-testable even by a two sided error adaptive algorithm making $o(n^{1/2})$ queries.*

Proof: By Claim 5.5 and Claim 5.6, given an ϵ-test of a 0/1 labeled $5n \times 5n$ matrix M for 'permutation-p', we construct a test of a 0/1/2 labeled $n\times n$ matrix A for 'permutation' by querying the location $(\lfloor \frac{i}{5} \rfloor, \lfloor \frac{i}{5} \rfloor)$ of A whenever the location (i,j) of M is queried, and assigning to (i,j) the entry of $G_{A(\lfloor\frac{i}{5}\rfloor,\lfloor\frac{i}{5}\rfloor)}$ at (i mod 5, j mod 5). In other words, the test of A is constructed by simulating the test of M on the tiling of A. The new test clearly makes the same number of queries as the original one.

We now claim that the new test is a 25ϵ-test for the property 'permutation'. In particular, this means that the existence of this $\frac{1}{500}$-test for 'permutation-p' would imply the existence of a corresponding $\frac{1}{20}$-test for 'permutation'. This would contradict Proposition 5.1, so the theorem follows.

Indeed, if A has 'permutation' then clearly its tiling M has 'permutation-p'. On the other hand let us assume that A is 25ϵ-far from satisfying 'permutation', and $\epsilon < \frac{1}{25}$. Claim 5.6 implies that any matrix M' that satisfies 'permutation-p' and is not ϵ-far from M is a tiling of some matrix A' that satisfies 'permutation'. This holds because one has to change M in at least $\frac{1}{25}(5n)^2$ places to remove all tight guide submatrices of the form $(M(5i+i', 5j+j'))_{0\le i',j'\le 4}$. Since A' is at least 25ϵ-far from A (by the assumption that A is far from any matrix that satisfies 'permutation'), the tiling M' of A' is at least ϵ-far from the tiling M of A. ■

6 Concluding remarks

We have seen that $\forall$-poset properties are testable, for 0/1 matrices as well as matrices over any fixed finite alphabet, while some $\forall\exists$-poset properties are not testable.

It is also interesting to investigate the 'submatrix' model, in which properties are defined by a set of forbidden submatrices, rather than forbidden posets. The situation with this model is not yet completely understood. [14] and [2] contain a relatively efficient test (with dependence on $1/\epsilon$ that is better then a tower) for the permutation invariant case using a 'conditional' Regularity Lemma that is proven there for the purpose. However, we do not know yet how to construct a tester for the case where permutation invariance is not guaranteed; many other interesting open questions also exist for the 'submatrix' model.

Back to the $\forall$-poset model: It would be nice to make the tests more efficient, especially in the case of non-binary alphabets. Another open problem is to better understand the $\exists\forall$-poset properties. This latter model is related to some colorability problem in the spirit of [11], and currently the question as to whether properties in this model are testable is open.

Finally, other interesting relations apart from the order relation can be used, giving rise to different models. For example, for the 2-dimensional case, $row(x,y)/\ col(x,y)$ that states that x and y are on the same row/ column. Others are $succ_R(x_1,x_2)$ stating that x_2 is on the same row as x_1 and directly at the right of x_1, and similarly $succ_C(x_1,x_2)$ for columns. The relations $row(x_1,x_2) \vee col(x_1,x_2)$ and $succ_R(x_1,x_2) \vee succ_C(x_1,x_2)$ are both expressible by "$\forall$" formulae using the basic poset-model relations, so "$\forall$" properties using them are all $\forall\exists$-poset properties according to the definition of Section 2 (but not necessarily $\forall$-poset properties). We currently have no results for these models.

Acknowledgment: We wish to thank the anonymous referee for the many helpful comments.

References

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

[2] N. Alon, E. Fischer and I. Newman, Testing of bipartite graph properties, in preparation.

[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):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 *RANDOM-APPROX99 Conference Proceedings*, pages 245–256, 1999.

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

[7] R. Diestel, *Graph Theory* (2^{nd} edition), Springer, 2000.

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

[9] F. Ergün, S. Kannan, S. R. Kumar, R. Rubinfeld and M. Viswanathan, Spot checkers, *JCSS*, 60(3):717–751, 2000.

[10] E. Fischer, On the strength of comparisons in property testing, *Information and Computation*, 189:107–116, 2004.

[11] E. Fischer, Testing graphs for colorability properties, In 12^{th} *SODA Conference Proceedings*, pages 873–882, 2001. Also: *Random Structures and Algorithms*, 26(3):289–309, 2005.

[12] 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.

[13] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld and A. Samorodnitsky, Monotonicity testing over general poset domains, In 34^{th} *ACM STOC Conference Proceedings*, pages 474–483, 2002.

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

[15] E. Fischer, I. Newman and J. Sgall, Functions that have read-twice constant width branching programs are not necessarily testable, *Random Structures and Algorithms* 24(2):175–193, 2004.

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

[17] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron and A. Samorodnitsky, Testing monotonicity, *Combinatorica*, 20(3):301–337, 2000.

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

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

[20] T. Kövary, V.T. Sós, and P. Turán, On a problem of K. Zarankiewicz, *Colloq. Math.*, 3:50–57, 1954.

[21] I. Newman, Testing membership in languages that have small width branching programs, *SIAM J. of Computing*, 31(5):1557–1570, 2002.

[22] 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), Vol. II, Chapter 15, Kluwer Press, 2001.

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

[24] E. Szemerédi, Regular partitions of graphs, In: *Proc. Colloque Inter. CNRS* No. 260 (J. C. Bermond, J. C. Fournier, M. Las Vergnas and D. Sotteau eds.), 2978, 399–401.

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

[26] K. Zarankiewicz, Problem P 101. *Colloq. Math.*, 2:116–131, 1951.