A linear lower bound for testing 3CNF properties

Sofya Raskhodnikova

Hebrew University of Jerusalem

(Supported by the Lady Davis Fellowship)

Joint work with Eli Ben-Sasson and Prahladh Harsha

Why property testing?

Property testing studies sublinear algorithms.

Goals

- Algorithmic techniques (like dynamic programming for P)
- Lower bound techniques (like NP-completeness for NP)

Motivation

- Massive data sets
  - scientific measurements, DNA sequences, WWW, census data, sales logs, high-resolution images
- Long access time
- Arises in the context of PCPs

What can sublinear algorithms do?

Can't even read the entire data!

- Some problems have exact deterministic sublinear solutions
- For most interesting problems algorithms must be
  - probabilistic
  - approximate

We know how to solve some problems:

- average, median values

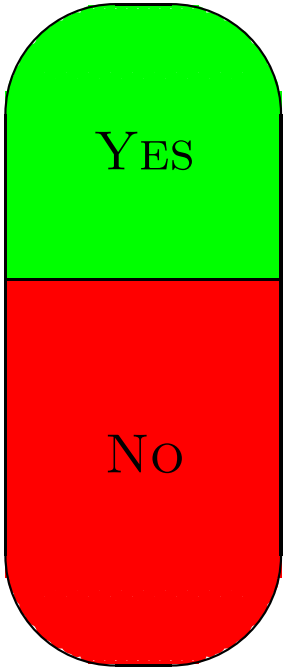
Types of approximation

- "Classical approximation"
  - output is close to the desired value.
- Property testing for decision problems
  - output is correct answer for given input, or at least some input close to it.

Property tester definition

[Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]

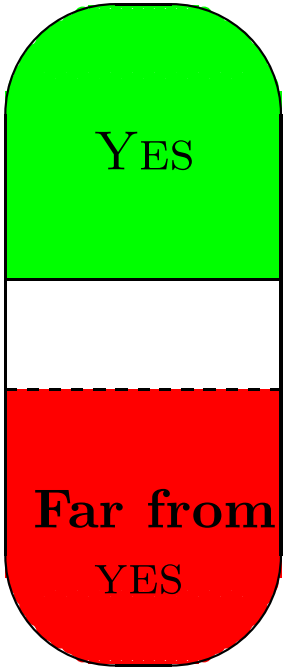
Probabilistic algorithm



Accept with high probability

Reject with high probability

Property tester



Accept with high probability

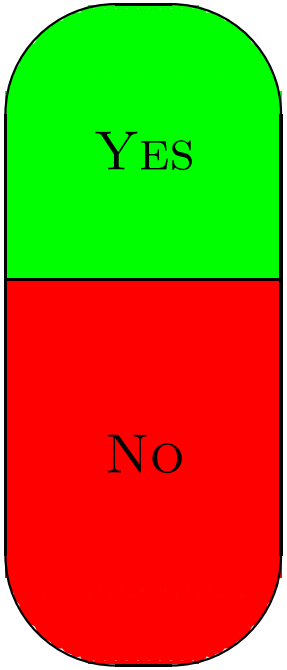
Do not care

Reject with high probability

Property tester definition

[Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]

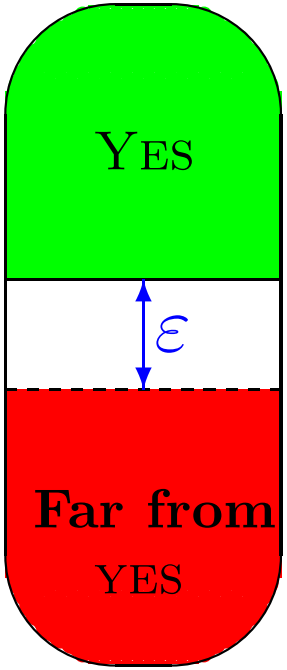
Probabilistic algorithm



Accept with high probability

Reject with high probability

Property tester



Accept with high probability

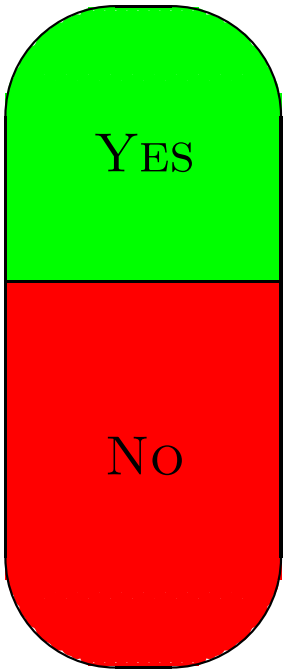
Do not care

Reject with high probability

Property tester definition

[Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98]

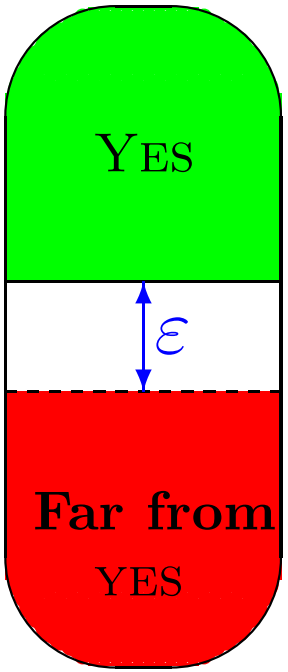
Probabilistic algorithm



Accept with high probability

Reject with high probability

Property tester



Accept with high probability

Do not care

Reject with high probability

Complexity measure: number of queries to the input.

1-sided error test always accepts positive instances.

Non-adaptive test makes all queries before getting any answers.

Testing world ≠ classical world

The approximate testing world is fascinating and different from the world of exact problems.

Problem	Exact	Testing
Is a 3CNF satisfiable? (3SAT)	NP-complete	easy [Alon Shapira 01]
Does an assignment satisfy a fixed 3CNF?	easy	hard [this talk]

Example: testing monotonicity

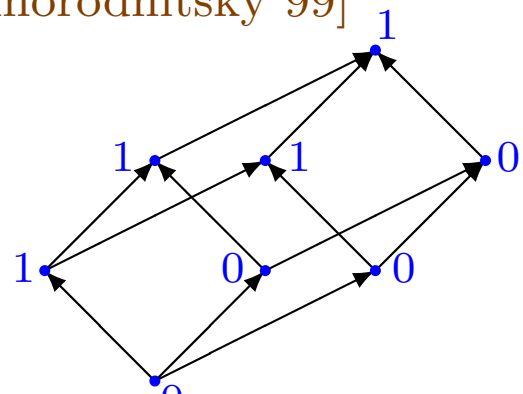
[Goldreich Goldwasser Lehman Ron 98,

Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

A function $f : \{0,1\}^m \to \{0,1\}$ is monotone if increasing a bit of x does not decrease $f(x)$.

Is f monotone?

Requires reading all 2^m values of f.



monotone

Example: testing monotonicity

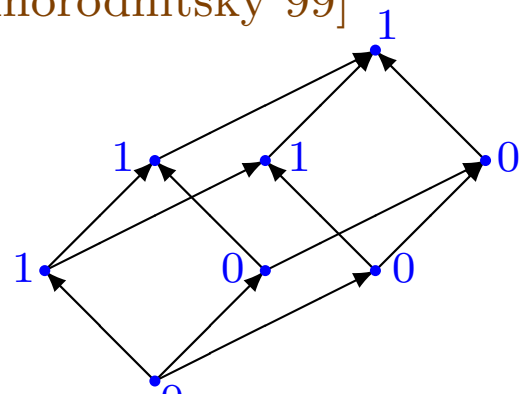
[Goldreich Goldwasser Lehman Ron 98,

Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

A function $f : \{0,1\}^m \to \{0,1\}$ is monotone if increasing a bit of x does not decrease $f(x)$.

Is f monotone?

Requires reading all 2^m values of f.

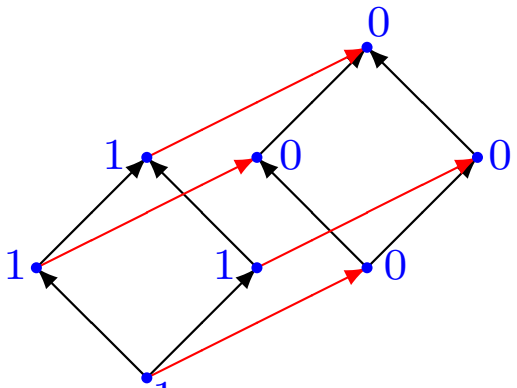


monotone

Is f monotone or ε-far from monotone (f has to change on many points to become monotone)?

Test

- Query endpoints of $\frac{m}{\varepsilon}$ random edges.
- Reject if an edge $1 \longrightarrow 0$ is found.



$\frac{1}{2}$-far from monotone

Example: testing triangle freeness

[Alon 01]

Is a graph triangle-free or ε-far from triangle-free?

Graph representation: Adjacency matrix.

ε-far: need to remove $\geq \varepsilon$ fraction of edges to kill all $\triangle$s

TEST

- Pick $t(\varepsilon)$ random triplets (v_1, v_2, v_3) of vertices.
- Query $(v_1, v_2), (v_2, v_3), (v_1, v_3)$ for each triplet.
- Reject if a triangle is discovered.

Representing properties by CNFs

- kCNF formula: AND of OR-clauses with at most k literals.
  E. g., $(x \vee \overline{y}) \wedge (y \vee \overline{z}) \wedge (x \vee z)$ is a 2CNF.
- Assume $\{0, 1\}$ alphabet.

Set of strings of length n that have the property:

000010...
011010...
110010...
...

x_1	x_2	x_3	...	x_n	has the property?
0	0	0	...	0	1
1	0	0	...	0	0
0	1	0	...	0	1
	.	.	.		

Write a Boolean formula for the property and convert it into CNF.

Our problem: kCNF testing

- Fixed: a CNF formula ϕ

  Input: an assignment to ϕ

  Determine: Is ϕ satisfied or ε-far from satisfied?

  (Need to change ε fraction of assignment to make ϕ satisfied).

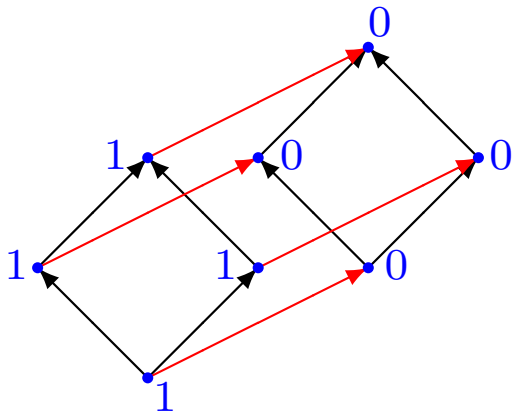
- Testers for properties describable by kCNF formulas.

- Do not confuse with: Given a kCNF formula, determine if it is satisfiable or ε-far from satisfiable. [Alon Shapira 01]

Examples of 2CNF and 3CNF properties

- Monotonicity is a 2CNF property:

$$x \longrightarrow y$$
$$\Updownarrow$$
$$x = 1 \text{ and } y = 0 \text{ is bad}$$
$$\Updownarrow$$
$$\overline{x} \vee y$$



- Triangle-freeness is a 3CNF property:

$$\text{triplet of vertices } (i, j, k)$$
$$\Updownarrow$$
$$\overline{e_{ij}} \vee \overline{e_{jk}} \vee \overline{e_{ik}}$$

A problem equivalent to 2CNF testing

Testing monotonicity of functions on PO domains

[Fischer Lehman Newman Raskhodnikova Rubinfeld Samorodnitsky 02]

Fixed: an acyclic directed graph

Input: a labeling of its nodes

Determine: if the labeling is monotone, i.e. if there are no violated edges along which labels decrease: $1 \longrightarrow 0$

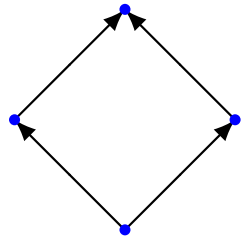
- 2CNF testing is equivalent monotonicity testing
  - 2CNF $\rightarrow$ implication graph

$O(\sqrt{n})$ *upper bound for monotonicity*

Let G be a transitively closed graph.

TEST

- Query $\Theta(\sqrt{\frac{n}{\varepsilon}})$ node labels at random.
- Reject if a violated edge is found in G.

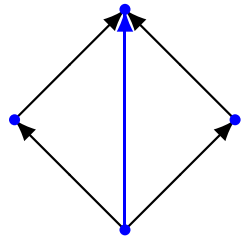


$O(\sqrt{n})$ *upper bound for monotonicity*

Let G be a transitively closed graph.

TEST

- Query $\Theta(\sqrt{\frac{n}{\varepsilon}})$ node labels at random.
- Reject if a violated edge is found in G.

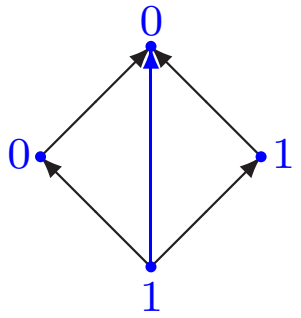


$O(\sqrt{n})$ *upper bound for monotonicity*

Let G be a transitively closed graph.

TEST

- Query $\Theta(\sqrt{\frac{n}{\varepsilon}})$ node labels at random.
- Reject if a violated edge is found in G.



If the labeling is monotone, the test accepts.

Suppose the labeling is ε-far from monotone.

- Let G' = graph of violated edges.

$$\text{distance to monotone} = |\text{min vertex cover}(G')|$$
$$|\,\text{max matching}(G')| \geq \tfrac{1}{2}|\,\text{min vertex cover}(G')| \geq \tfrac{1}{2}\varepsilon n$$

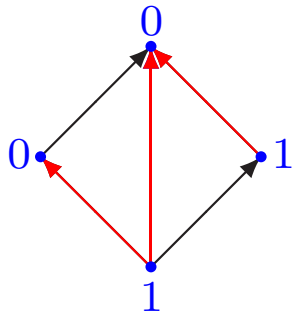
- By Birthday Paradox-like argument, the test is likely to query two endpoints of an edge in the max matching.

$O(\sqrt{n})$ *upper bound for monotonicity*

Let G be a transitively closed graph.

TEST

- Query $\Theta(\sqrt{\frac{n}{\varepsilon}})$ node labels at random.
- Reject if a violated edge is found in G.



If the labeling is monotone, the test accepts.

Suppose the labeling is ε-far from monotone.

- Let G' = graph of violated edges.

$$\text{distance to monotone} = |\text{min vertex cover}(G')|$$
$$|\,\text{max matching}(G')| \geq \tfrac{1}{2}|\,\text{min vertex cover}(G')| \geq \tfrac{1}{2}\varepsilon n$$

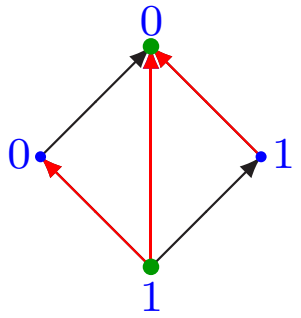
- By Birthday Paradox-like argument, the test is likely to query two endpoints of an edge in the max matching.

$O(\sqrt{n})$ *upper bound for monotonicity*

Let G be a transitively closed graph.

TEST

- Query $\Theta(\sqrt{\frac{n}{\varepsilon}})$ node labels at random.
- Reject if a violated edge is found in G.



If the labeling is monotone, the test accepts.

Suppose the labeling is ε-far from monotone.

- Let G' = graph of violated edges.

$$\text{distance to monotone} = |\text{min vertex cover}(G')|$$
$$|\,\text{max matching}(G')| \geq \tfrac{1}{2}|\,\text{min vertex cover}(G')| \geq \tfrac{1}{2}\varepsilon n$$

- By Birthday Paradox-like argument, the test is likely to query two endpoints of an edge in the max matching.

Testing hierarchy?

- Testing 2CNFs can be done with $O(\sqrt{n})$ queries.
- How about kCNF?
  - "witness" of size k that the formula is falsified
  - Monotone kCNFs (with no negations) can be tested with $O(n^{\frac{k-1}{k}})$ queries.

Testing hierarchy?

- Testing 2CNFs can be done with $O(\sqrt{n})$ queries.
- How about kCNF?
  - "witness" of size k that the formula is falsified
  - Monotone kCNFs (with no negations) can be tested with $O(n^{\frac{k-1}{k}})$ queries.

Main Theorem Some 3CNF properties require a linear number of queries.

Non-testable properties

- Non-testable properties exist [GGR98]

- Explicit linear lower bounds
  - 3-colorable graphs in bounded degree model
    [Bogdanov Obata Trevisan 02]
  - Degree $n/2$ polynomials over $\mathbb{F}_n$
    represented by values on all points [Sudan]

Non-testable properties

- Non-testable properties exist [GGR98]

  $\Downarrow$

  Some CNF properties are non-testable.

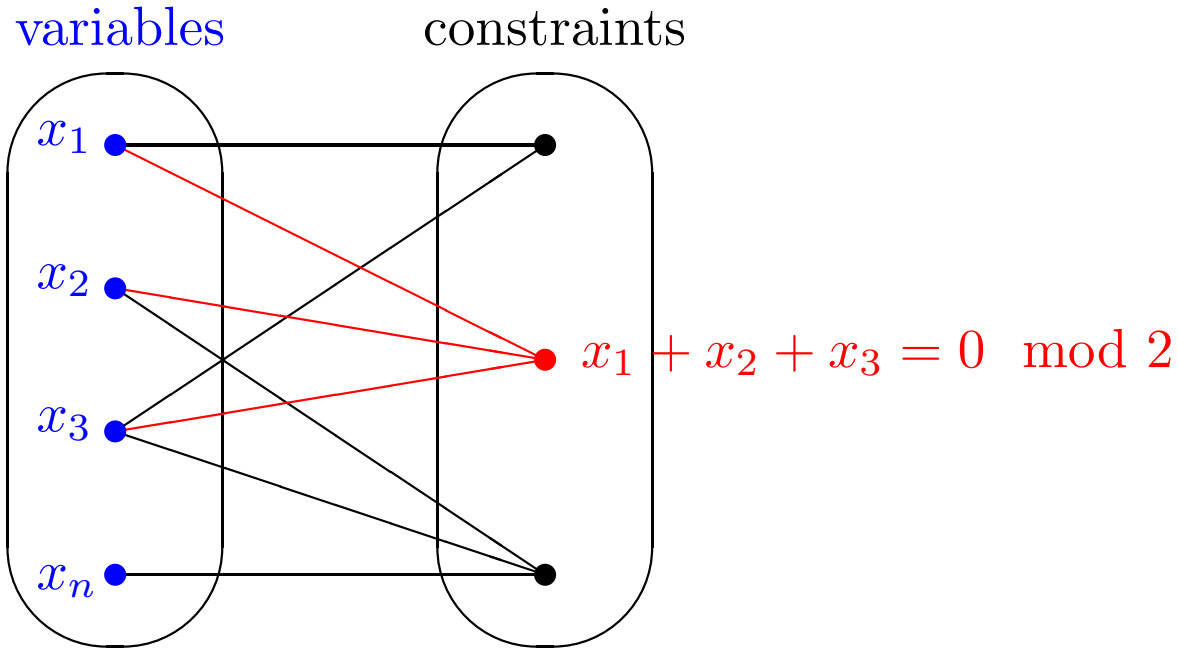
- Explicit linear lower bounds
  - 3-colorable graphs in bounded degree model

    [Bogdanov Obata Trevisan 02]
  - Degree $n/2$ polynomials over $\mathbb{F}_n$
    represented by values on all points [Sudan]
  - No small witnesses showing violations

Testing 3CNFs is hard

Main Theorem Some 3CNF properties require a linear number of queries.

Idea for hard 3CNF properties: Use vector spaces



V is set of strings in $\{0,1\}^n$ satisfying all constraints.

V is a vector space.

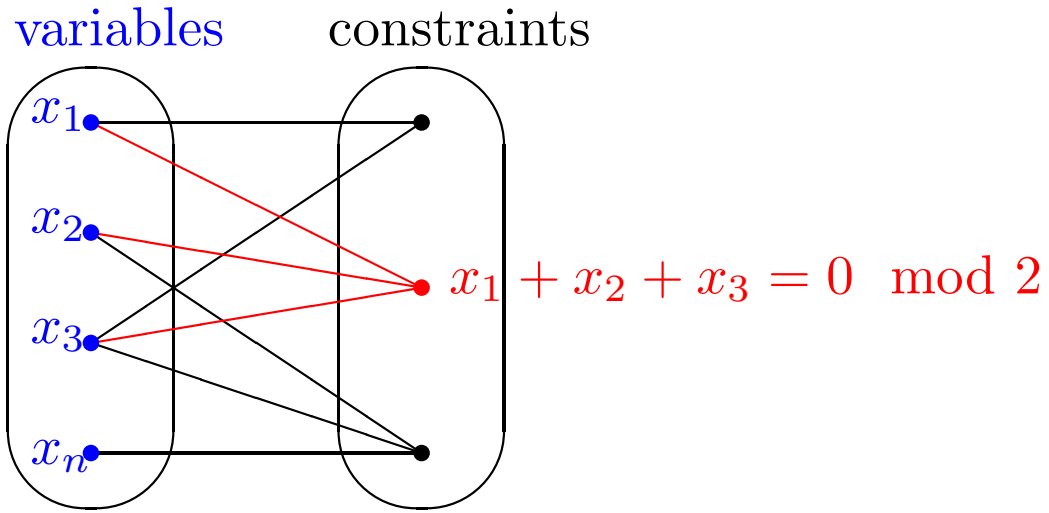
If right degree $\leq d$, V is a dCNF property.

Outline of the proof

Some 3CNF properties require a linear number of queries.

- Sufficient conditions for V to be hard
  for 1-sided error non-adaptive tests.
- Random LDPC (c, d)-codes satisfy the conditions
  (for fixed $d > 3$).
- Reducing d to 3.
- Dealing with adaptivity and 2-sided error:
  - For linear properties (vector spaces),
    adaptivity and having 2-sided error do not help.

1-sided error tests



V is the space of vectors satisfying all constraints.
$V^\perp$ is the space generated by the constraints.

- 1-sided error test rejects only if it detects a violation:

no string in V is consistent with the answers

$\Updownarrow$ linearity

some α in $V^\perp$ contains only queried variables and is falsified by the answers

Yao's principle for 1-sided error tests

Goal: Find hard to test vector spaces V.

Prove they are hard to test by playing a game.

Yao's principle for 1-sided error tests

Goal: Find hard to test vector spaces V.

Prove they are hard to test by playing a game.

Players:

evil algorithms designer Bill G. and poor lower bound prover Alice

Game 1
Move 1 Bill selects a q-query probabilistic test.
Move 2 Alice selects a bad (far from V) input vector which the test is likely to accept.

Game 2
Move 1 Alice selects an input distribution on bad vectors.
Move 2 Bill selects a q-query deterministic test which is likely to reject a vector from the distribution.

Yao's principle: Alice can perform in game 1 at least as well as in game 2.

Fooling 1-sided error non-adaptive tests

Given a graph and a corresponding vector space V,
how do we prove V is hard to test?

Yao's principle: Enough to give bad distribution B on vectors x s.t.

1. All x under B are ε-far from V.
2. $\Pr_{x \leftarrow B}[T \text{ rejects } x]$ is low for every q-query test T.

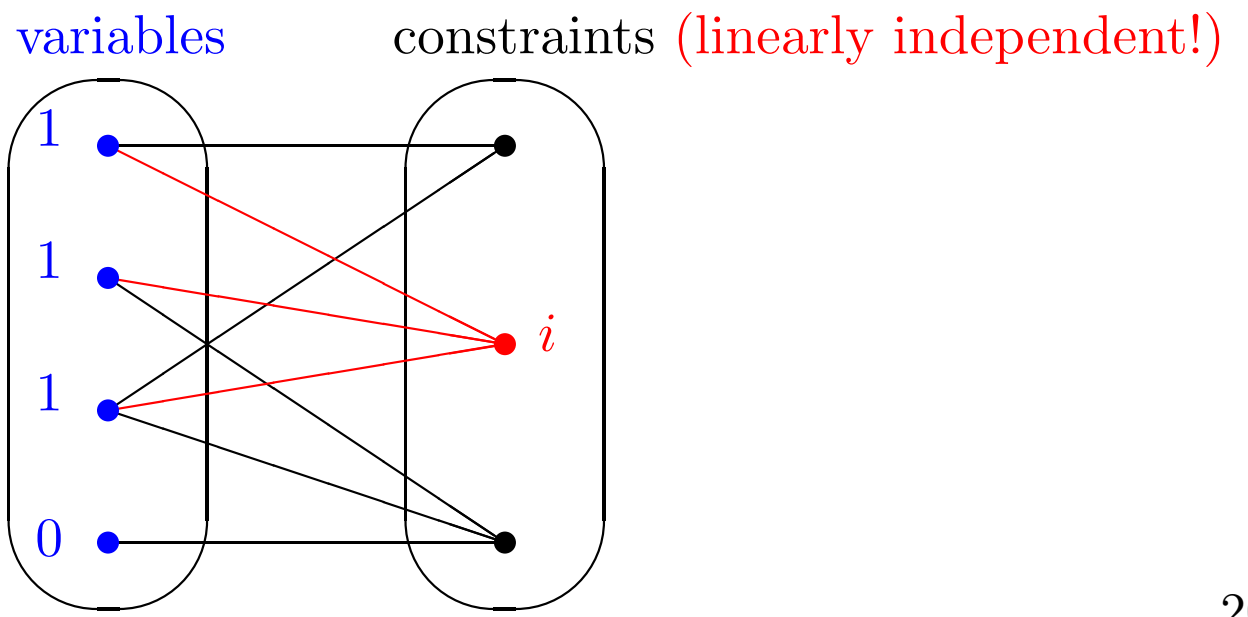
Fooling 1-sided error non-adaptive tests

Given a graph and a corresponding vector space V,
how do we prove V is hard to test?

Yao's principle: Enough to give bad distribution B on vectors x s.t.

1. All x under B are ε-far from V.
2. $\Pr_{x \leftarrow B}[T \text{ rejects } x]$ is low for every q-query test T.

Distribution B: uniformly pick a constraint i;
uniformly pick a vector that falsifies only i.



Conditions on V implying that B is bad

A linearly independent set of constraints is

1. ε-separating: $\forall x$ that falsify exactly one constraint, $|x| \geq \varepsilon n$.

Conditions on V implying that B is bad

A linearly independent set of constraints is

1. ε-separating: $\forall x$ that falsify exactly one constraint, $|x| \geq \varepsilon n$.

$$\Downarrow$$

All x under B are ε-far from V.

Proof: $\forall x \leftarrow B, y \in V$, $x+y$ falsifies exactly one constraint.
$\text{dist}(x, y) = |x + y| \geq \varepsilon n$.

Two representations of each constraint

Each constraint in $V^\perp$ has two representations:

1. a vector $\alpha \in \{0,1\}^n$ indicating the variables in the constraint
2. a sum of basis constraints

Let $c_1, \ldots, c_m$ be the basis constraints.

$$\alpha = \begin{pmatrix} | & | & & | \\ c_1 & c_2 & \cdots & c_m \\ | & | & & | \end{pmatrix} u \quad \text{where} \begin{cases} u \in \{0,1\}^m \\ u_i = 1 \text{ if } c_i \text{ is in the sum} \end{cases}$$

Conditions on V implying that B is bad

A linearly independent set of constraints is

1. ε-separating: $\forall x$ that falsify exactly one constraint, $|x| \geq \varepsilon n$.
2. (q, μ)-local: Every constraint in $V^{\perp}$ of weight $\leq q$ is a sum of $\leq \mu$ fraction of basis constraints.

Conditions on V implying that B is bad

A linearly independent set of constraints is

1. ε-separating: $\forall x$ that falsify exactly one constraint, $|x| \geq \varepsilon n$.
2. (q, μ)-local: Every constraint in $V^{\perp}$ of weight $\leq q$ is a sum of $\leq \mu$ fraction of basis constraints.

$$\Downarrow$$

$$\Pr_{x \leftarrow B}[T \text{ rejects } x] < 2\mu \text{ for every } q\text{-query test } T.$$

Intuition: If a constraint α has weight $\leq q$, querying variables in α gives a q-query test.

- The probability that this test finds a violation under B is the fraction of basis constraints in the representation of α.

(q, μ)-*locality*

2. (q, μ)-local: Every constraint in $V^{\perp}$ of weight $\leq q$ is a sum of $\leq \mu$ fraction of basis constraints.

$$\Downarrow$$

$$\Pr_{x \leftarrow B}[T \text{ rejects } x] < 2\mu \text{ for every } q\text{-query test } T.$$

Proof: Let Q be the set of queries asked by T.

- T rejects only if some $\alpha \in V^{\perp}$ has support in Q and α's basis representation contains the violated constraint c_i:

$$\alpha = \begin{pmatrix} | & | & & | \\ c_1 & c_2 & \cdots & c_m \\ | & | & & | \end{pmatrix} u \quad \text{where } u_i = 1$$

- How many coordinates can be 1 in at least one such vector u?
  - such vectors u form a vector space $U \in \{0,1\}^m$
  - (q, μ)-locality $\implies |u| \leq \mu m$

A combinatorial lemma

Suppose all vectors in a vector space U in $\{0,1\}^m$ have weight $\leq w$.
Let k = number of coordinates that are 1 in at least one vector.
How big can k be?

- If x_i is not identically 0, it is 1 in half of the vectors.
- The number of appearances of 1 over all vectors:

$$k\frac{|U|}{2} = \sum_{u \in U} |u| \leq (|U| - 1)w < |U|w$$

$$k < 2w$$

Outline of the proof

Some 3CNF properties require a linear number of queries.

- Sufficient conditions for V to be hard
  for 1-sided error non-adaptive tests.

- Random LDPC (c, d)-codes satisfy the conditions
  (for fixed $d > 3$).
- Reducing d to 3.
- Dealing with adaptivity and 2-sided error:
  - For linear properties (vector spaces),
    adaptivity and having 2-sided error do not help.

Outline of the proof

Some 3CNF properties require a linear number of queries.

- Sufficient conditions for V to be hard
  for 1-sided error non-adaptive tests.
- Random LDPC (c, d)-codes satisfy the conditions
  (for fixed $d > 3$).
- Reducing d to 3.
- Dealing with adaptivity and 2-sided error:
  - For linear properties (vector spaces),
    adaptivity and having 2-sided error do not help.

Reducing d to 3 in d-linear formulas

Reduction: ith constraint $x_1 + \cdots + x_d = 0$ is replaced by

$x_1 + \cdots + x_{\lceil \frac{d}{2} \rceil} + z_i = 0$ and $x_{\lceil \frac{d}{2} \rceil + 1} + \cdots + x_d + z_i = 0$.

- The reduction is repeated $\log d$ times.
- Needed conditions hold with slightly worse parameters.

Reducing d to 3 in d-linear formulas

Reduction: ith constraint $x_1 + \cdots + x_d = 0$ is replaced by

$x_1 + \cdots + x_{\lceil \frac{d}{2} \rceil} + z_i = 0$ and $x_{\lceil \frac{d}{2} \rceil + 1} + \cdots + x_d + z_i = 0$.

- The reduction is repeated $\log d$ times.
- Needed conditions hold with slightly worse parameters.

E.g., new constraints are linearly independent.

$$\Updownarrow$$

No set C of new constraints sums up to 0.

– If all constraints in C come in pairs,

new sum = sum of the corresponding old constraints $\neq 0$

– If some lonely constraint corresponding to an old constraint i appears in C, the sum has 1 in ith position.

Outline of the proof

Some 3CNF properties require a linear number of queries.

- Sufficient conditions for V to be hard
  for 1-sided error non-adaptive tests.
- Random LDPC (c, d)-codes satisfy the conditions
  (for fixed $d > 3$).
- Reducing d to 3.
- Dealing with adaptivity and 2-sided error:
  - For linear properties (vector spaces),
    adaptivity and having 2-sided error do not help.

Removing 2-sided error and adaptivity

$(\varepsilon, \mu_+, \mu_-, q)$-test for V

- makes q queries
- accepts all x in V with probability $\geq 1 - \mu_+$
- accepts all x ε-far from V with probability $\leq \mu_-$

Removing 2-sided error and adaptivity

$(\varepsilon, \mu_+, \mu_-, q)$-test for V

- makes q queries
- accepts all x in V with probability $\geq 1 - \mu_+$
- accepts all x ε-far from V with probability $\leq \mu_-$

For a linear space $V \subseteq \mathbb{F}^n$

2-sided error adaptive $(\varepsilon, \mu_+, \mu_-, q)$-test for V

$\Downarrow$

1-sided error adaptive $(\varepsilon, 0, \mu_+ + \mu_-, q)$-test for V

$\Downarrow$

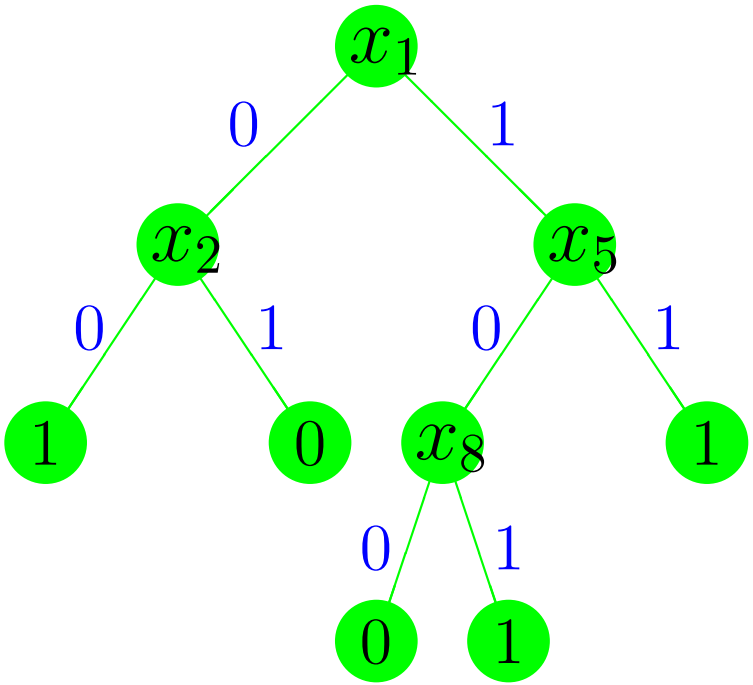
1-sided error non-adaptive $(\varepsilon, 0, \mu_+ + \mu_-, q)$-test for V

General tests

probabilistic adaptive test

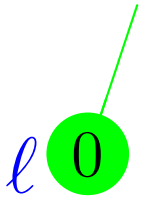
=

distribution on decision trees



Useful facts

Distance: Vectors in $V + x$ have the same distance to V.



V_ℓ = vectors in V that reach leaf ℓ.

V_ℓ^x = vectors in $V + x$ that reach leaf ℓ.

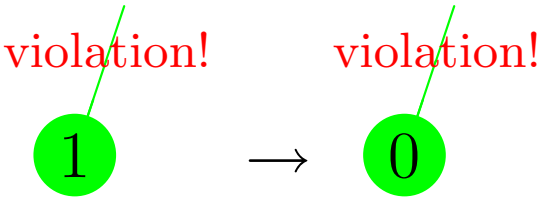
Leaf Lemma. If both V_ℓ and V_ℓ^x are non-empty,

$$|V_\ell| = |V_\ell^x|.$$

Removing 2-sided error

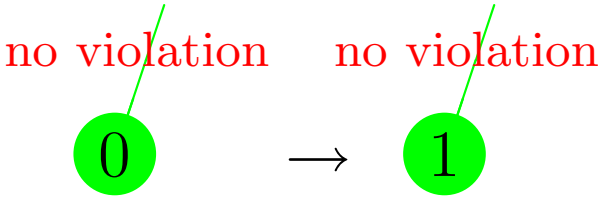
Relabel leaves optimally: leaf ℓ is 0 iff

answers on the path to ℓ falsify a constraint.



$x \in V$: $\Pr[T(x) = 1]$ doesn't change

bad x: $\Pr[T(x) = 1]$ can only go down



$x \in V$: $\Pr[T(x) = 1]$ can only go up

bad x: $\Pr[T(x) = 1]$ can go up, too!

Removing 2-sided error

Relabel leaves optimally: leaf ℓ is 0 iff

answers on the path to ℓ falsify a constraint.

violation! violation!

1 → 0

$x \in V$: $\Pr[T(x) = 1]$ doesn't change

bad x: $\Pr[T(x) = 1]$ can only go down

no violation no violation

0 → 1

$x \in V$: $\Pr[T(x) = 1]$ can only go up

bad x: $\Pr[T(x) = 1]$ can go up, too!

"On average", the relabeling is good.

Removing 2-sided error: new test

New test T'

On input x,

- Pick a random $v \in V$. Output $T_{\text{opt}}(x+v)$.

Notation: $\rho^x = average_{y \in V+x} \Pr[T(y)=1]$

$\rho^{\bar{0}} = \rho$

Claim For all $x \notin V$, relabeling does not decrease $\rho - \rho^x$.

Then for all bad x $\Pr[T'(x)=1] = \rho^x_{\text{opt}}$

$\leq \rho^x + \rho_{\text{opt}} - \rho$

$\leq \mu_- + 1 - (1-\mu_+)$

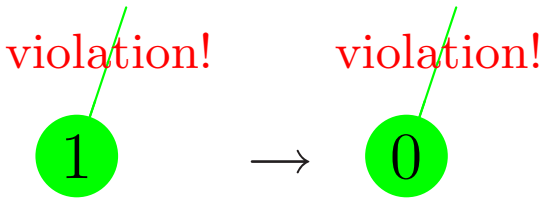
$= \mu_- + \mu_+$

Proof of the claim

Notation: $\rho^x = average_{y \in V+x} \Pr[T(y) = 1]$

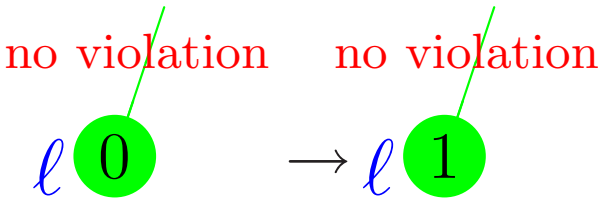
$\rho^{\bar{0}} = \rho$

Claim For all $x \notin V$, relabeling does not decrease $\rho - \rho^x$.



ρ doesn't change

ρ^x can only go down



ρ^x increases at most as much as ρ

$\Pr[T(x) = 1]$ changes only for x that reach ℓ.

Need to show $|V_\ell| \geq |V_\ell^x|$.

By Leaf Lemma, $|V_\ell| = |V_\ell^x|$ if both $V_\ell \neq \emptyset$ and $V_\ell^x \neq \emptyset$.

Path leading to ℓ does not falsify constraints $\Rightarrow V_\ell \neq \emptyset$.

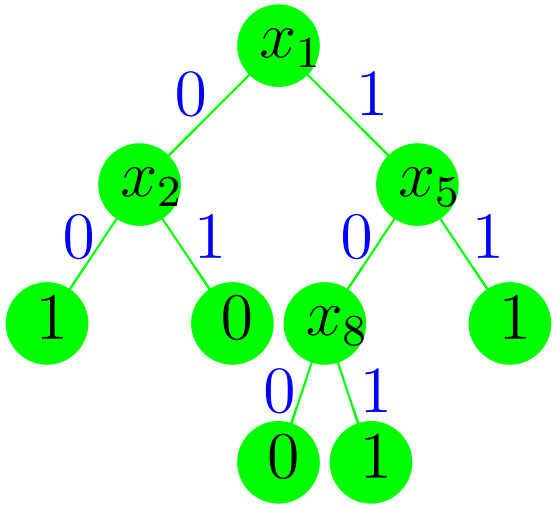
Removing adaptivity

New test T''

On input x,

- Pick a random $v \in V$.
- Query x on variables queried by T_{opt} on input v.
- Reject iff a violation is detected.

Claim For all x, $\Pr[T''(x) = 1] = \rho^x_{\text{opt}}$.



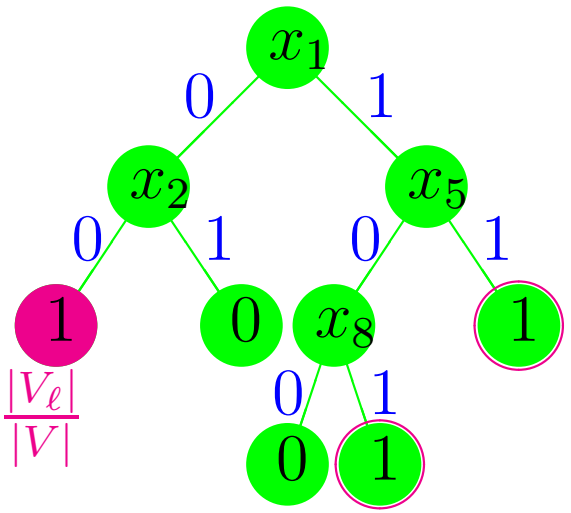
Removing adaptivity

New test T''

On input x,

- Pick a random $v \in V$.
- Query x on variables queried by T_{opt} on input v.
- Reject iff a violation is detected.

Claim For all x, $\Pr[T''(x) = 1] = \rho^x_{\text{opt}}$.



Outline of the proof

Some 3CNF properties require a linear number of queries.

- Sufficient conditions for V to be hard
  for 1-sided error non-adaptive tests.
- Random LDPC (c, d)-codes satisfy the conditions
  (for fixed $d > 3$).
- Reducing d to 3.
- Dealing with adaptivity and 2-sided error:
  - For linear properties (vector spaces),
    adaptivity and having 2-sided error do not help.

Open questions

- Techniques for adaptive lower bounds for non-linear properties (not vector spaces)?
- Lower bounds on testing codes with linear distance and constant rate?
- Better characterization of testable properties?