## LOW-DEGREE TESTING (version 1)

**Input** Table of some function $f: \mathbb{F}_2^n \to \mathbb{F}_2$
(or $f: \{0,1\}^n \to \{0,1\}$ — identify $\mathbb{F}_2 = \{0,1\}$)

**Problem** Decide whether $f$ is a polynomial in $\mathbb{F}_2[y_1, \ldots, y_n]$ of total degree $\leq r$

**Constraint** Run very fast
Ideally in time $O(1)$ in terms of $n$
(Running time might depend on $r$)

Two observations

(1) Every $f: \mathbb{F}_2^n \to \mathbb{F}_2$ is a polynomial, so the only question is what the degree is

Define for $a \in \mathbb{F}_2$

$$\chi_a(x) = \begin{cases} 1-x & \text{if } a=0 \\ x & \text{if } a=1 \end{cases}$$

For $\vec{a} \in \mathbb{F}_2^n$

$$\chi_{\vec{a}}(\vec{x}) = \prod_{i=1}^{n} \chi_{a_i}(x_i)$$

Clearly $\chi_{\vec{a}}$ degree-$n$ polynomial such that

$$\chi_{\vec{a}}(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} = \vec{a} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$f(\vec{x}) = \sum_{\vec{a} \in \mathbb{F}_2^n} f(\vec{a}) \cdot \chi_{\vec{a}}(\vec{x})$$

(2) Problem as posed above, impossible! II

Suppose we have low-degree testing algorithm A
Check what positions $U \subseteq \mathbb{F}_2^n$ are read by A
(will say A QUERIES these positions)

$|U| \ll 2^n$, so algorithm can't detect if we corrupt positions outside of U

Have to allow RANDOMNESS

Still doesn't help. Consider function table of $\chi_{\vec{a}}(\vec{x})$ for some fixed $\vec{a} \in \mathbb{F}_2^n$.

Even randomized A will see $\vec{a}$ with probability $2^{-\Omega(n)}$. Apart from $\vec{a}$, function looks like zero polynomial $\Rightarrow$ low degree
But has degree n!

But $\chi_{\vec{a}}(\vec{x})$ is zero polynomial "except for noise" in point $\vec{a}$.

Low-degree testing algorithm:
- Should be allowed to use randomness
- Should determine whether f is a low-degree polynomial **except possibly** for noise.

The DISTANCE between f and g is

$$\delta(f,g) = \Pr_{\vec{x} \in \mathbb{F}_2^n}\left[f(\vec{x}) + g(\vec{x})\right] = \frac{\left|\{\vec{x} \in \mathbb{F}_2^n : f(\vec{x}) + g(\vec{x})\}\right|}{\left|\mathbb{F}_2^n\right|}$$

Distance from class of functions $\mathcal{G}$

$$\delta(f,\mathcal{G}) = \min_{g \in \mathcal{G}}\{\delta(f,g)\}$$

Let $P_r = \{ g \in \mathbb{F}_2[x_1, \ldots, x_n] \mid \deg(g) \leq r \}$ III

LOW-DEGREE TESTING (version 2)

Input: Table of $f : \mathbb{F}_2^n \to \mathbb{F}_2$, parameters $r, \varepsilon$

Constant: Query $f$ in constant number of positions (i.e., independent of $n$, but can depend on $r$ and $\varepsilon$)

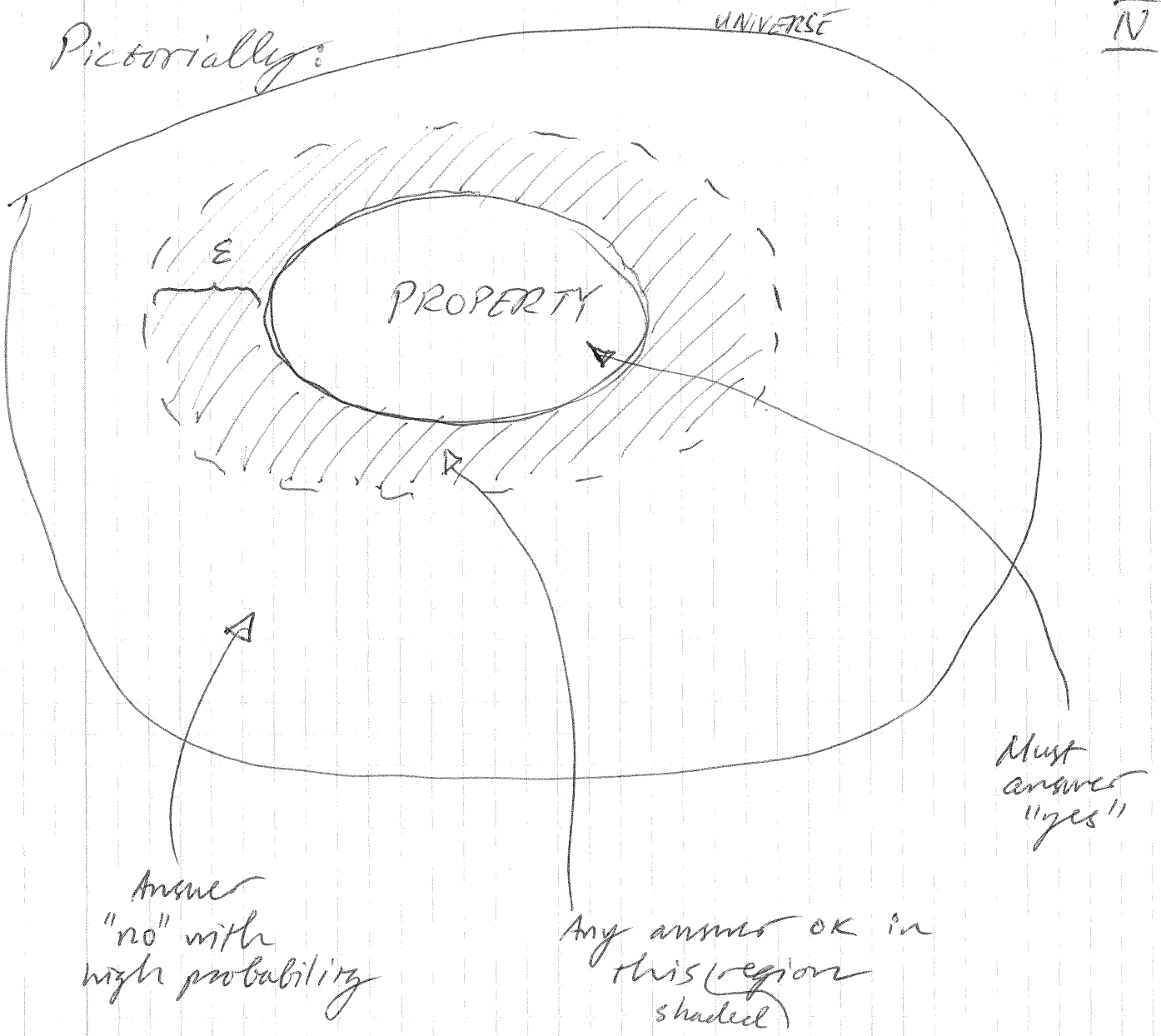
Task: (a) If $\deg(f) \leq r$ (i.e., $f \in P_r$), output YES (always)

(b) If $\delta(f, P_r) \geq \varepsilon$, output NO with probability at least $2/3$ (probability over internal randomness, not over input)

Comments:

- An algorithm $A$ solving this problem is a PROPERTY TESTER with PERFECT COMPLETENESS (no error if $f \in P_r$) and SOUNDNESS ERROR $1/3$, also known as a ONE-SIDED (PROPERTY) TESTER.
- The exact constant $2/3$ ($1/3$) is arbitrary — can go from any constant to any other (smaller) constant by repeating test.
- This is a PROMISE PROBLEM. If neither (a) or (b) applies, then algorithm can give any answer (so we are "promised" that either (a) or (b) applies).

Pictorially:




- There is an implicit parameterization in terms of $n$, which is usually ignored in the notation. That is, we have

$$\mathcal{P}_r^n = \{ g \in \mathbb{F}_2[x_1, \ldots, x_n] \mid g \text{ } n\text{-variate}, \deg(g) \leq r \}$$

and

$$\mathcal{P}_r = \bigcup_{n=1}^{\infty} \mathcal{P}_r^n$$

All asymptotic statements are w.r.t $n \to \infty$

From now on, we will follow the custom to suppress $n$ in the notation

WHY CARE?

- This kind of property testing is an entire area of research in TCS with many fascinating results.

- Can be viewed as extreme form of decoding-like question for error-correcting codes.
  Reed-Muller codewords are evaluations of low-degree polynomials. We are asking whether RM codes are LOCALLY TESTABLE (check in sublinear time whether received word is a codeword or far from all codewords).
  Other flavours of this problem:
  - LOCALLY DECODABLE CODES – decode individual positions in message in sublinear time
  - LOCALLY CORRECTABLE CODES – retrieve individual positions in codeword correctly (even if this particular position was corrupted in transit)

- Very important for hardness of approximation
  Low-degree testing at the heart of the PCP Theorem
  ([Arora, Lund, Motwani, Sudan, & Szegedy '98] and other papers)
  saying, roughly, that any NP-witness (for satisfiability of a CNF formula, say) can be encoded in such a way that it can be checked by randomly querying a constant number of positions.

- Finally, seems like a fairly natural and interesting question about polynomials. And in this course, we care about polynomials, so we care, in particular, about low-degree testing. :-)

---

TODAY AND NEXT LECTURE:

Discuss result that provides almost optimal tester for low-degree polynomials

Alon, Kaufman, Krivelevich, Litsyn, & Ron
"Testing Reed-Muller Codes"
IEEE Transactions on Information Theory 2005
(Conference version in APPROX-RANDOM 2003)

Two ways to view testing algorithm

(1) Code theoretic view

We want to test Reed-Muller code $C$

Look at dual code $C^\perp$ $\{\vec{x} \in \mathbb{F}_2^n \mid \langle \vec{x}, \vec{c} \rangle = 0 \ \forall \vec{c} \in C\}$ ← also Reed-Muller code

Given received word $y$

Pick random codewords $x_i$ of minimum (constant) Hamming weight for $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k$, $k = O_n(1)$.

Accept $y$ if $\forall \vec{x}_i \quad \langle \vec{x}_i, \vec{y} \rangle = 0$

Works if $\vec{y} \in \mathbb{F}_2^n$ if s.t. $\Delta(\vec{y}, C) \geq \varepsilon n$ has high probability of having $\langle \vec{x}_i, \vec{y} \rangle \neq 0$ for random min-weight $\vec{x}_i \in C^\perp$

(Will not use this view from now on.)

(2) Polynomial view

Pick random subspace $S$ of dimension $r+1$ in $\mathbb{F}_2^n$

Check if $f$ restricted to $S$ is degree-$r$ polynomial — accept if yes, reject otherwise

For historic reasons, [AKKLR05] focuses on degree-$r$ polynomials with constant term 0, i.e., $f(\vec{0}) = 0$

So we will do so as well. Let

$$P_r^* = \{ g \in \mathbb{F}_2[x_1, \ldots, x_n] \mid \deg(g) \le r,\ g(\vec{0}) = 0 \}$$

We will solve low-degree testing problem for $P_r^*$ instead. Difference is not very important.

A BRIEF HISTORY

Blum-Luby-Rubinfeld '93

Testing of Hadamard codes

= linear functions over $\mathbb{F}_2^n$ = degree-1 polynomials

Test:

Repeat a suitable number of times:

Pick $\vec{y}_1, \vec{y}_2 \in \mathbb{F}_2^n$ uniformly and independently at random

Reject if $f(\vec{y}_1) + f(\vec{y}_2) \neq f(\vec{y}_1 + \vec{y}_2)$

Accept if all tests passed.

This test clearly accepts all linear functions.

Less obviously, test is good at detecting functions that are far from being linear, as shown in [BLR93]

(Improved analysis in [Bellare, Coppersmith, Håstad, Kiwi, & Sudan '96] using discrete Fourier analysis.)

Also well-studied for fields $\mathbb{F}_q$ of characteristic $q >$ degree bound $r$

General idea:

- Pick random line $L$
- Check if $f|_L$ ($f$ restricted to $L$) is univariate degree-$r$ poly
- Checking $r+1$ points on $L$ tells us what $p(t) = f|_L$ should be; now evaluate at some other point to check consistency

This tests illustrates why we should expect dependence on $r$ in # queries — $f$ evaluated on too few points always looks consistent with some polynomial of degree $\leq r$.

Unfortunately, this idea does not work for $\mathbb{F}_2$

Instead, generalize BLR test:

AKKLR-TESTER$(f, r, \varepsilon)$

Repeat $\Theta\left(\frac{1}{\varepsilon \cdot 2^r} + r \cdot 2^r\right)$ times:

- Uniformly and independently at random choose $\vec{y}_1, \ldots, \vec{y}_{r+1} \in \mathbb{F}_2^n$
- Evaluate $f$ on all nontrivial linear combinations of the $\vec{y}_i$ and reject if

$$\sum_{\emptyset \neq I \subseteq [r+1]} f\left(\sum_{i \in I} \vec{y}_i\right) \neq 0 \qquad (1)$$

Accept if all tests passed

THEOREM 1 [AKKLR 05]

The AKKLR-tester:

(a) has query complexity $O\left(\frac{1}{\varepsilon} + r \cdot 2^{2r}\right)$

(b) always accepts $f \in P_r^*$

(c) rejects $f$ that are $\varepsilon$-far from $P_r^*$ with probability $\geq 2/3$ (over internal randomness of algorithm).

---

We have to prove

- completeness (b) – usually not too hard
- soundness (c) – usually tricky part

Claim (a) about # evaluations of $f$ will follow from closer description

Since we are going to write the basic AKKLR-test in (1) many times, we want some notation

For $f: \mathbb{F}_2^n \to \mathbb{F}_2$ and $\vec{y}_1, \ldots, \vec{y}_\ell \in \mathbb{F}_2^n$, let $T_f(\vec{y}_1, \ldots, \vec{y}_\ell)$ denote sum of $f$ evaluated at all points in subspace $\operatorname{span}(\vec{y}_1, \ldots, \vec{y}_\ell)$ except $\vec{0}$, i.e.,

$$T_f(\vec{y}_1, \ldots, \vec{y}_\ell) = \sum_{\emptyset \neq I \subseteq [\ell]} f\left(\sum_{i \in I} \vec{y}_i\right) \qquad (2)$$

**LEMMA 2** (Completeness)

$f \in \mathcal{P}_r^*$ if and only if for all $\vec{y}_1, \ldots, \vec{y}_{r+1} \in \mathbb{F}_2^n$ it holds that $T_f(\vec{y}_1, \ldots, \vec{y}_{r+1}) = 0$.

*Proof*: Need to show

($\Rightarrow$) $f \in \mathcal{P}_r^* \Rightarrow T_f(\vec{y}_1, \ldots, \vec{y}_{r+1}) = 0$

Sufficient to prove for monomial $m$ of degree $\leq r$

($\Leftarrow$) Write $f(\vec{x}) = \sum_{I \subseteq [n]} a_I \prod_{i \in I} x_i$

Show that $a_\emptyset = 0$ and $a_I = 0$ for $|I| > r$ if $f$ satisfies $T_f(\vec{y}_1, \ldots, \vec{y}_{r+1}) = 0 \ \forall \vec{y}_1, \ldots, \vec{y}_{r+1}$.

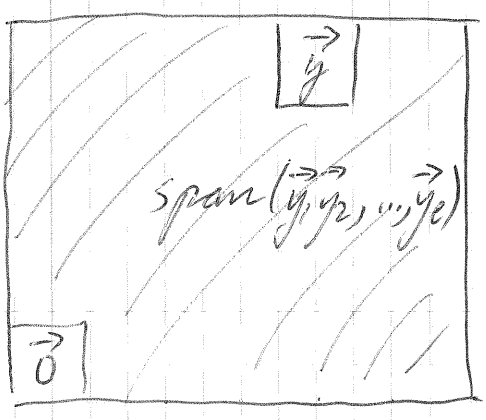
Detailed proof left as an exercise. ▨

Let us also define

$$T_f^{\vec{y}}(\vec{y}_2, \ldots, \vec{y}_\ell) = T_f(\vec{y}, \vec{y}_2, \ldots, \vec{y}_\ell) + f(\vec{y}) \quad (3)$$

$T_f^{\vec{y}}$ evaluates $f$ everywhere on subspace $\mathrm{span}(\vec{y}, \vec{y}_2, \ldots, \vec{y}_\ell)$ except $\vec{0}$ and $\vec{y}$ (since $f(\vec{y})$-evaluations cancel in $\mathbb{F}_2$).

Or in pictures...




In view of Lem 2, if $f$ is a degree-$r$ poly then $T_f^{\vec{y}}$ should be a way to evaluate $f(\vec{y})$ without looking at $\vec{y}$.

OBSERVATION 3

If $f \in P_r^*$, then for all $\vec{y}_2, \ldots, \vec{y}_{r+1} \in \mathbb{F}_2^n$ it holds that $T_f^{\vec{y}}(\vec{y}_2, \ldots, \vec{y}_{r+1}) = f(\vec{y})$

Proof Adding $f(\vec{y})$ to both sides yields $T_f(\vec{y}, \vec{y}_2, \ldots, \vec{y}_{r+1}) = 0$, which holds if $f$ is a degree-$r$ poly with constant term 0 by Lem 2 ☐

It remains to prove soundness, i.e. that if $f$ is far from being a degree-$r$ poly, then AKKLR-tester rejects with (sufficiently) high probability.

Usually proven in contrapositive direction: Suppose that tester for property $P$ accepts $f$ with "too high" probability. Then $f$ is inside the "shaded region" in our figure, i.e., there is a $g \in P$ such that the distance $\delta(f, g)$ is small.

This is how the AKKLR proof goes as well

---

Let us suppose that $f$ is indeed close to some polynomial $g$

Which polynomial should this be? How can we recover it?

Idea: Take "majority vote"

Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be defined by

$$g(\vec{y}) = \begin{cases} 1 & \text{if } \Pr_{\vec{y}_2, \ldots, \vec{y}_{r+1} \sim \mathbb{F}_2^n}\left[T_f^{\vec{y}}(\vec{y}_2, \ldots, \vec{y}_{r+1}) = 1\right] \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\vec{y}_i \sim \mathbb{F}_2^n$ denotes uniformly random and independent samples.

That is, if we sample a random subspace containing $\vec{y}$ and evaluate everywhere except $\vec{0}$ and $\vec{y}$, then $g(\vec{y})$ is majority vote on what value at $\vec{y}$ should be to satisfy local constraint that we have a degree-$r$ poly.

Define also

$$\gamma = \Pr_{\vec{y}_1, \ldots, \vec{y}_{r+1} \sim \mathbb{F}_2^n}\left[T_f(\vec{y}_1, \ldots, \vec{y}_{r+1}) \neq 0\right] \quad (5)$$

to be probability that single iteration of AKKLR-tester detects that $f$ is not a degree-$r$ polynomial

OBSERVATION 4

$$\gamma = \Pr_{\vec{y}, \vec{y}_2, \ldots, \vec{y}_{r+1} \sim \mathbb{F}_2^n}\left[T_f^{\vec{y}}(\vec{y}_2, \ldots, \vec{y}_{r+1}) \neq f(\vec{y})\right]$$

Proof Just substitute (3) into (5).

Now the rest of the proof goes roughly as follows:

Suppose that (one iteration of AKKLR-tester rejects $f$ with probability $\gamma \geq \varepsilon$.

If so, tester is doing the right thing — $\gamma > 0$ only if $f$ is not degree-$r$ poly

Furthermore, repeating test $\sim 1/\varepsilon$ times will boost reject probability to constant (which can then be further boosted to $2/3$).

Suppose, however, that $\gamma \ll \varepsilon$. Then we want to find "excuse" for AKKLR tester to be reluctant to reject, i.e., show that $f$ is close to some degree-$r$ poly

This part divides into two claims

(1) $f$ is close to $g$ as defined in (4)

(2) If $\gamma$ is small enough, then $g$ is a degree-$r$ polynomial.

claim (1) is relatively easy.

claim (2) is substantially more work (will be the focus of next lecture).

LEMMA 5 (f and g are close)

Let $f: \mathbb{F}_2^n \to \mathbb{F}_2$ be any function, and let $g: \mathbb{F}_2^n \to \mathbb{F}_2$ be defined by (3) and $\gamma$ by (5). Then $f$ and $g$ are $2\gamma$-close, i.e., $\delta(f,g) \leq 2\gamma$.

Proof Consider subset $U \subseteq \mathbb{F}_2^n$ such that for all $\vec{y} \in U$ it holds that

$$\Pr_{\vec{y}_2,\ldots,\vec{y}_{r+1}}\left[ f(\vec{y}) \neq T_f^{\vec{y}}(\vec{y}_2,\ldots,\vec{y}_{r+1}) \right] > \frac{1}{2}.$$

Details are left as an exercise. ▨

The roadmap for the rest of the proof is now as follows.

First, by definition of $g$ we know that for at least half of the $r$-tuples $\vec{y}_2,\ldots,\vec{y}_{r+1} \in \mathbb{F}_2^n$ it holds that

$$g(\vec{y}) = T_f^{\vec{y}}(\vec{y}_2,\ldots,\vec{y}_{r+1})$$

But if $\gamma$ is small, then this holds in fact for a vast majority of the $r$-tuples, not just half of them.

LEMMA 6

For every $\vec{y} \in \mathbb{F}_2^n$ it holds that

$$\Pr_{\vec{y}_2,\ldots,\vec{y}_{r+1}}\left[ g(\vec{y}) = T_f^{\vec{y}}(\vec{y}_2,\ldots,\vec{y}_{r+1}) \right] \geq 1 - 2r\gamma$$

then we will prove that if detection probability $\eta$ of one iteration of AKKLR-test is small enough, then $g$ is a low-degree poly

LEMMA 7

If $\eta < \frac{1}{(4r+2)2^r}$, then $g \in \mathcal{P}^*_r$.

This would already give us a tester if we are willing to do $\Omega(1/\varepsilon)$ iterations. But we are shooting for only $\Omega(1/(2^r \cdot \varepsilon))$ iterations.

For this we need:

LEMMA 8

Suppose that $0 < \eta < \frac{1}{(4r+2)2^r}$. Then for $\vec{y}_1, \ldots, \vec{y}_{r+1}$ chosen uniformly and independently at random, the probability that exactly one point $\vec{v} \in \left\{ \sum_{i \in I} \vec{y}_i \mid \emptyset \neq I \subseteq [r+1] \right\}$ satisfies $f(\vec{v}) \neq g(\vec{v})$ is at least

$$\frac{1}{3}\left(2^{r+1} - 1\right) \delta(f,g)$$

COROLLARY 9

If $\eta < \frac{1}{(4r+2)2^r}$, then $\eta \geq \frac{1}{3}\left(2^{r+1}-1\right)\delta(f,g)$

Proof $g$ is a polynomial, so $T^{f-g}(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_{r+1}) = 0$ always. If $f(\vec{v}) \neq g(\vec{v})$ in exactly one point, $T^{f-g}(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_{r+1}) \neq 0$ and the AKKLR-tester rejects $f$. ☐

We can now prove the main theorem, which we restate for reference

THEOREM 1 [AKKLR05]

The AKKLR-tester

(a) has query complexity $O\left(\frac{1}{\varepsilon} \cdot r 2^{2r}\right)$

(b) always accepts $f \in P_r^*$

(c) rejects $f$ s.t. $\delta(f, P_r^*) \geq \varepsilon$ with probability $\geq 2/3$

Proof: If $f \in P_r^*$, then AKKLR-tester accepts by Lemma 2.

Suppose that $\delta(f, P_r^*) \geq \varepsilon$.

If we run AKKLR-tester for $O(1/\eta)$ iterations, then tester will reject with any constant probability we choose. (Expected # iterations needed $= 1/\eta$, and if we do $C \cdot 1/\eta$ iterations then probability of detection failure tails off like $\exp(-C)$.)

So it is sufficient to show that

$$\eta = \Omega\left(\min\left(2^r \cdot \varepsilon, \frac{1}{r \cdot 2^r}\right)\right).$$

If $\eta \geq \frac{1}{(4r+2)2^r}$ we're done, so suppose

$$\eta < \frac{1}{(4r+2)2^r}$$

By (Lem 8 and) Cor 9, we have

$$\gamma \geq \frac{1}{3}(2^{r+1}-1)\ \delta(f,g)$$

By Lemma 5 we have $g \in P_r^*$. Since $\delta(f, P_r^*) \geq \varepsilon$, it follows that

$$\delta(f,g) \geq \varepsilon$$

Hence

$$\gamma \geq \frac{1}{3}(2^{r+1}-1)\cdot\varepsilon = \Omega(2^r\cdot\varepsilon)$$

and the theorem follows ∎

Oh, and # iterations

$$O\left(\frac{1}{2^r\cdot\varepsilon} + r\cdot 2^r\right)$$

plus $2^{r+1}-1$ evaluations per iteration yields bound on query complexity