Read about Algebraic probabilistic algorithm.
(polynomial)

Online homepage

Blackground reading : polynomial

Polynomial identity testing :

$P(x) = C_n x^n + C_{n-1} x^{n-1} + \cdots + C_1 x^1 + C_0$

coefficients from a field : Ex : rations, real, complex
Z_p, prime

Operation : $+ -$ 2 poly
$\times \div$ 2 poly
evaluate poly

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

Polynomial identity testing

From Wikipedia, the free encyclopedia

In mathematics, **polynomial identity testing** (PIT) is the problem of efficiently determining whether two multivariate polynomials are identical. More formally, a PIT algorithm is given an arithmetic circuit that computes a polynomial p in a field, and decides whether p is the zero polynomial. Determining the computational complexity required for polynomial identity testing is one of the most important open problems in algebraic computing complexity.

$P_1(x) - P_2(x) = 0$
$\Leftrightarrow$ all coefficients are 0. $\rightarrow$ not efficient
$\Leftrightarrow \forall x [P_1(x) - P_2(x) = 0]$. eg. $P_1(x) = (3x-2)^{50}$

Description [edit]

The question "Does $(x+y)(x-y)$ equal $x^2 - y^2$?" is a question about whether two polynomials are identical. As with any polynomial identity testing question, it can be trivially transformed into the question "Is a certain polynomial equal to 0?"; in this case we can ask "Does $(x+y)(x-y) - (x^2 - y^2) = 0$"? If we are given the polynomial as an algebraic expression (rather than as a black-box), we can confirm that the equality holds through brute-force multiplication and addition, but the time complexity of the brute-force approach grows as $\binom{n+d}{d}$, where n is the number of variables (here, $n = 2$: x is the first and y is the second), and d is the degree of the polynomial (here, $d = 2$). If n and d are both large, $\binom{n+d}{d}$ grows exponentially.[1]

PIT concerns whether a polynomial is identical to the zero polynomial, rather than whether the function implemented by the polynomial always evaluates to zero in the given domain. For example, the field with two elements, GF(2), contains only the elements 0 and 1. In GF(2), $x^2 - x$ always evaluates to zero; despite this, PIT does not consider $x^2 - x$ to be equal to the zero polynomial.[2]

Determining the computational complexity required for polynomial identity testing is one of the most important open problems in the mathematical subfield known as "algebraic computing complexity".[1][3] The study of PIT is a building-block to many other areas of computational complexity, such as the proof that IP=PSPACE.[1][4] In addition, PIT has applications to Tutte matrices and also to primality testing, where PIT techniques led to the AKS primality test, the first deterministic (though impractical) polynomial time algorithm for primality testing.[1]

Formal problem statement [edit]

Given an arithmetic circuit that computes a polynomial in a field, determine whether the polynomial is equal to the zero polynomial (that is, the polynomial with no nonzero terms).[1]

Solutions [edit]

In some cases, the specification of the arithmetic circuit is not given to the PIT solver, and the PIT solver can only input values into a "black box" that implements the circuit, and then analyze the output. Note that the solutions below assume that any operation (such as multiplication) in the given field takes constant time; further, all black-box algorithms below assume the size of the field is larger than the degree of the polynomial.

The Schwartz–Zippel algorithm provides a practical probabilistic solution, by simply randomly testing inputs and checking whether the output is zero. It was the first randomized polynomial time PIT algorithm to be proven correct.[1] The larger the domain the inputs are drawn from, the less likely Schwartz–Zippel is to fail. If random bits are in short supply, the Chen-Kao algorithm (over the rationals) or the Lewin-Vadhan algorithm (over any field) require fewer random bits at the cost of more required runtime.[2]

A *sparse PIT* has at most m nonzero monomial terms. A sparse PIT can be deterministically solved in polynomial time of the size of the circuit and the number m of monomials,[1] see also.[5]

A *low degree PIT* has an upper bound on the degree of the polynomial. Any low degree PIT problem can be reduced in subexponential time of the size of the circuit to a PIT problem for depth-four circuits; therefore, PIT for circuits of depth-four (and below) is intensely studied.[1]

See also [edit]

- Applications of Schwartz–Zippel lemma

External links [edit]

- Lecture notes on "Polynomial Identity Testing by the Schwartz-Zippel Lemma"
- Polynomial Identity Testing by Michael Forbes - MIT on YouTube
- Prize winner for Polynomial Identity Testing

References [edit]

1. ^ a b c d e f g h Saxena, Nitin. "Progress on Polynomial Identity Testing." Bulletin of the EATCS 99 (2009): 49-79.
2. ^ a b Shpilka, Amir, and Amir Yehudayoff. "Arithmetic circuits: A survey of recent results and open questions." Foundations and Trends in Theoretical Computer Science 5.3–4 (2010): 207-388.
3. ^ Dvir, Zeev, and Amir Shpilka. "Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits." SIAM Journal on Computing 36.5 (2007): 1404-1434.
4. ^ Adi Shamir. "IP=PSPACE." Journal of the ACM (JACM) 39.4 (1992): 869-877.
5. ^ Grigoriev,Dima, Karpinski,Marek, and Singer,Michael F., "Fast Parallel Algorithms for Sparse Multivariate Polynomial Interpolation over Finite Fields", SIAM J. Comput., Vol 19, No.6, pp. 1059-1063, December 1990

Categories: Polynomials | Computer algebra

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

In other projects
Wikibooks

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Español
Français
한국어
हिन्दी
Italiano
Русский
Tiếng Việt
中文
46 more
Edit links

Field (mathematics)

From Wikipedia, the free encyclopedia

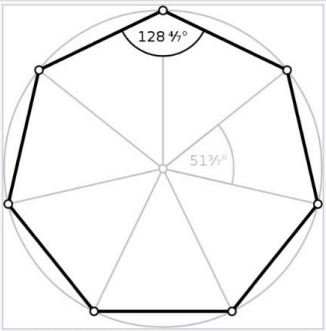
This article is about fields in algebra. For fields in geometry, see Vector field. For other uses, see Field § Mathematics.

In mathematics, a **field** is a set on which addition, subtraction, multiplication, and division are defined and behave as the corresponding operations on rational and real numbers do. A field is thus a fundamental algebraic structure which is widely used in algebra, number theory, and many other areas of mathematics.

The best known fields are the field of rational numbers, the field of real numbers and the field of complex numbers. Many other fields, such as fields of rational functions, algebraic function fields, algebraic number fields, and *p*-adic fields are commonly used and studied in mathematics, particularly in number theory and algebraic geometry. Most cryptographic protocols rely on finite fields, i.e., fields with finitely many elements.

The relation of two fields is expressed by the notion of a field extension. Galois theory, initiated by Évariste Galois in the 1830s, is devoted to understanding the symmetries of field extensions. Among other results, this theory shows that angle trisection and squaring the circle cannot be done with a compass and straightedge. Moreover, it shows that quintic equations are algebraically unsolvable.

Fields serve as foundational notions in several mathematical domains. This includes different branches of analysis, which are based on fields with additional structure. Basic theorems in analysis hinge on the structural properties of the field of real numbers. Most importantly for algebraic purposes, any field may be used as the scalars for a vector space, which is the standard general context for linear algebra. Number fields, the siblings of the field of rational numbers, are studied in depth in number theory. Function fields can help describe properties of geometric objects.



The regular heptagon cannot be constructed using only a straightedge and compass construction; this can be proven using the field of constructible numbers.

Algebraic structures

Group-like	[show]
Ring-like	[show]
Lattice-like	[show]
Module-like	[show]
Algebra-like	[show]

V · T · E

Contents [hide]

Schwartz–Zippel lemma

From Wikipedia, the free encyclopedia

In mathematics, the **Schwartz–Zippel lemma** (also called the **DeMillo-Lipton-Schwartz–Zippel lemma**) is a tool commonly used in probabilistic polynomial identity testing, i.e. in the problem of determining whether a given multivariate polynomial is the 0-polynomial (or identically equal to 0). It was discovered independently by Jack Schwartz,[1] Richard Zippel,[2] and Richard DeMillo and Richard J. Lipton, although DeMillo and Lipton's version was shown a year prior to Schwartz and Zippel's result.[3] The finite field version of this bound was proved by Øystein Ore in 1922.[4]

Statement of the lemma [edit]

The input to the problem is an n-variable polynomial over a field $\mathbf{F}$. It can occur in the following forms:

Algebraic form [edit]

For example, is

$$(x_1 + 3x_2 - x_3)(3x_1 + 4z_4 - 1)\cdots(z_7 - z_2) \equiv 0\,?$$

To solve this, we can multiply it out and check that all the coefficients are 0. However, this takes exponential time. In general, a polynomial can be algebraically represented by an arithmetic formula or circuit.

Determinant of a matrix with polynomial entries [edit]

Let

$$\boldsymbol{p(x_1, x_2, \ldots, x_n)}$$

be the determinant of the polynomial matrix.

Currently, there is no known sub-exponential time algorithm that can solve this problem deterministically. However, there are randomized polynomial algorithms for testing polynomial identities. Their analysis usually requires a bound on the probability that a non-zero polynomial will have roots at randomly selected test points. The Schwartz–Zippel lemma gives this as follows:

Theorem 1 (Schwartz, Zippel). *Let*

$$\boldsymbol{P \in F[x_1, x_2, \ldots, x_n]}$$

be a non-zero polynomial of total degree $d \geq 0$ over a field F. Let S be a finite subset of F and let $r_1, r_2, \ldots, r_n$ be selected at random independently and uniformly from S. Then

$$\Pr[P(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}.$$

In the single variable case, this follows directly from the fact that a polynomial of degree d can have no more than d roots. It seems logical, then, to think that a similar statement would hold for multivariable polynomials. This is, in fact, the case.

Proof. The proof is by mathematical induction on n. For $n = 1$, as was mentioned before, P can have at most d roots. This gives us the base case. Now, assume that the theorem holds for all polynomials in $n - 1$ variables. We can then consider P to be a polynomial in x_1 by writing it as

$$P(x_1, \ldots, x_n) = \sum_{i=0}^{d} x_1^i P_i(x_2, \ldots, x_n).$$

Since P is not identically 0, there is some i such that P_i is not identically 0. Take the largest such i. Then $\deg P_i \leq d - i$, since the degree of $x_1^i P_i$ is at most d.

Now we randomly pick $r_2, \ldots, r_n$ from S. By the induction hypothesis, $\Pr[P_i(r_2, \ldots, r_n) = 0] \leq \frac{d-i}{|S|}$.

If $P_i(r_2, \ldots, r_n) \neq 0$, then $P(x_1, r_2, \ldots, r_n)$ is of degree i (and thus not identically zero) so

$$\Pr[P(r_1, r_2, \ldots, r_n) = 0 | P_i(r_2, \ldots, r_n) \neq 0] \leq \frac{i}{|S|}.$$

If we denote the event $P(r_1, r_2, \ldots, r_n) = 0$ by A, the event $P_i(r_2, \ldots, r_n) = 0$ by B, and the complement of B by B^C, we have

$$\begin{aligned}\Pr[A] &= \Pr[A \cap B] + \Pr[A \cap B^C] \\ &= \Pr[B]\Pr[A|B] + \Pr[B^C]\Pr[A|B^C] \\ &\leq \Pr[B] + \Pr[A|B^C] \\ &\leq \frac{d-i}{|S|} + \frac{i}{|S|} = \frac{d}{|S|}\end{aligned}$$

Applications [edit]

The importance of the Schwartz–Zippel Theorem and Testing Polynomial Identities follows from algorithms which are obtained to problems that can be reduced to the problem of polynomial identity testing.

Comparison of two polynomials [edit]

Given a pair of polynomials $\boldsymbol{p_1(x)}$ and $\boldsymbol{p_2(x)}$, is

$$\boldsymbol{p_1(x) \equiv p_2(x)}\,?$$

This problem can be solved by reducing it to the problem of polynomial identity testing. It is equivalent to checking if

$$\boldsymbol{[p_1(x) - p_2(x)] \equiv 0.}$$

Hence if we can determine that

$$\boldsymbol{p(x) \equiv 0,}$$

where

$$\boldsymbol{p(x) = p_1(x) - p_2(x),}$$

then we can determine whether the two polynomials are equivalent.

Comparison of polynomials has applications for branching programs (also called binary decision diagrams). A read-once branching program can be represented by a multilinear polynomial which computes (over any field) on {0,1}-inputs the same Boolean function as the branching program, and two branching programs compute the same function if and only if the corresponding polynomials are equal. Thus, identity of Boolean functions computed by read-once branching programs can be reduced to polynomial identity testing.

Comparison of two polynomials (and therefore testing polynomial identities) also has applications in 2D-compression, where the problem of finding the equality of two 2D-texts A and B is reduced to the problem of comparing equality of two polynomials $\boldsymbol{p_A(x, y)}$ and $\boldsymbol{p_B(x, y)}$.

Primality testing [edit]

Given $\boldsymbol{n \in \mathbb{Z}^+}$, is $\boldsymbol{n}$ a prime number?

A simple randomized algorithm developed by Manindra Agrawal and Somenath Biswas can determine probabilistically whether $\boldsymbol{n}$ is prime and uses polynomial identity testing to do so.

They propose that all prime numbers n (and only prime numbers) satisfy the following polynomial identity:

$$\boldsymbol{(1 + z)^n = 1 + z^n (\mathrm{mod}\ n).}$$

This is a consequence of the Frobenius endomorphism.

Let

$$\boldsymbol{\mathcal{P}_n(z) = (1 + z)^n - 1 - z^n.}$$

Then $\boldsymbol{\mathcal{P}_n(z) = 0\ (\mathrm{mod}\ n)}$ *iff n is prime.* The proof can be found in [4]. However, since this polynomial has degree $\boldsymbol{n}$, and since $\boldsymbol{n}$ may or may not be a prime, the Schwartz–Zippel method would not work. Agrawal and Biswas use a more sophisticated technique, which divides $\boldsymbol{\mathcal{P}_n}$ by a random monic polynomial of small degree.

Prime numbers are used in a number of applications such as hash table sizing, pseudorandom number generators and in key generation for cryptography. Therefore, finding very large prime numbers (on the order of (at least) $\boldsymbol{10^{500} \approx 2^{1024}}$) becomes very important and efficient primality testing algorithms are required.

Perfect matching [edit]

Let $\boldsymbol{G = (V, E)}$ be a graph of n vertices where n is even. Does G contain a perfect matching?

Theorem 2 (Tutte 1947): *A Tutte matrix determinant is not a 0-polynomial if and only if there exists a perfect matching.*

A subset D of E is called a matching if each vertex in V is incident with at most one edge in D. A matching is perfect if each vertex in V has exactly one edge that is incident to it in D. Create a *Tutte matrix* A in the following way:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

where

$$\boldsymbol{a_{ij}} = \begin{cases} x_{ij} & \text{if } (i,j) \in E \text{ and } i < j \\ -x_{ji} & \text{if } (i,j) \in E \text{ and } i > j \\ 0 & \text{otherwise.} \end{cases}$$

The Tutte matrix determinant (in the variables x_{ij}, $\boldsymbol{i < j}$) is then defined as the determinant of this skew-symmetric matrix which coincides with the square of the pfaffian of the matrix A and is non-zero (as polynomial) if and only if a perfect matching exists. One can then use polynomial identity testing to find whether G contains a perfect matching. There exists a deterministic black-box algorithm for graphs with polynomially bounded permanents (Grigoriev & Karpinski 1987).[5]

In the special case of a balanced bipartite graph on $\boldsymbol{n = m + m}$ vertices this matrix takes the form of a block matrix

$$\boldsymbol{A} = \begin{pmatrix} 0 & X \\ -X^t & 0 \end{pmatrix}$$

if the first m rows (resp. columns) are indexed with the first subset of the bipartition and the last m rows with the complementary subset. In this case the pfaffian coincides with the usual determinant of the $m \times m$ matrix X (up to sign). Here X is the Edmonds matrix.

Notes [edit]

1. ^ (Schwartz 1980)
2. ^ (Zippel 1979)
3. ^ (DeMillo & Lipton 1978)
4. ^ Ø. Ore, Über höhere Kongruenzen. Norsk Mat. Forenings Skrifter Ser. I (1922), no. 7, 15 pages.
5. ^ (Grigoriev & Karpinski 1987)

References [edit]

- Agrawal, Manindra; Biswas, Somenath (2003-02-21). "Primality and Identity Testing via Chinese Remaindering" (PDF). *Journal of the ACM*. **50** (4): 429–443. doi:10.1145/792538.792540. Retrieved 2008-06-15.
- Berman, Piotr; Karpinski, Marek; Larmore, Lawrence L.; Plandowski, Wojciech; Rytter, Wojciech (2002). "On the Complexity of Pattern Matching for Highly Compressed Two-Dimensional Texts" (ps). *Journal of Computer and System Sciences*. **65** (2): 332–350. doi:10.1006/jcss.2002.1828. Retrieved 2008-06-15.
- Grigoriev, Dima; Karpinski, Marek (1987). *The matching problem for bipartite graphs with polynomially bounded permanents is in NC. Proceedings of the Annual Symposium on Foundations of Computer Science*. pp. 166–172. doi:10.1109/SFCS.1987.55. ISBN 978-0-8186-0807-0.
- Moshkovitz, Dana (2010). An Alternative Proof of The Schwartz-Zippel Lemma. ECCC TR10-096.
- DeMillo, Richard A.; Lipton, Richard J. (1978). "A probabilistic remark on algebraic program testing". *Information Processing Letters*. **7** (4): 193–195. doi:10.1016/0020-0190(78)90067-4.
- Rudich, Steven (2004). *AMS* (ed.). *Computational Complexity Theory*. IAS/Park City Mathematics Series. **10**. ISBN 978-0-8218-2872-4.
- Schwartz, Jack (October 1980). "Fast probabilistic algorithms for verification of polynomial identities" (PDF). *Journal of the ACM*. **27** (4): 701–717. CiteSeerX 10.1.1.391.1254. doi:10.1145/322217.322225. Retrieved 2008-06-15.
- Tutte, W.T. (April 1947). "The factorization of linear graphs" (PDF). *J. London Math. Soc.* **22** (2): 107–111. doi:10.1112/jlms/s1-22.2.107. hdl:10338.dmlcz/128615. Retrieved 2008-06-15.
- Zippel, Richard (1979). *Probabilistic algorithms for sparse polynomials. Lecture Notes in Computer Science*. **72**. pp. 216–226. doi:10.1007/3-540-09519-5_73. ISBN 978-3-540-09519-4.
- Zippel, Richard (February 1989). "An Explicit Separation of Relativized Random Polynomial Time and Relativized Deterministic Polynomial Time" (ps). Retrieved 2008-06-15.
- Zippel, Richard (1993). Springer (ed.). *Effective Polynomial Computation*. The Springer International Series in Engineering and Computer Science. **241**. ISBN 978-0-7923-9375-7.

External links [edit]

- "The Curious History of the Schwartz–Zippel Lemma", by Richard J. Lipton

Categories: Polynomials | Computer algebra | Lemmas | Mathematical theorems in theoretical computer science

Polynomial matrix

From Wikipedia, the free encyclopedia

Not to be confused with matrix polynomial.

In mathematics, a **polynomial matrix** or **matrix of polynomials** is a matrix whose elements are univariate or multivariate polynomials. Equivalently, a polynomial matrix is a polynomial whose coefficients are matrices.

A univariate polynomial matrix P of degree p is defined as:

$$P = \sum_{n=0}^{p} A(n)x^n = A(0) + A(1)x + A(2)x^2 + \cdots + A(p)x^p$$

where $A(i)$ denotes a matrix of constant coefficients, and $A(p)$ is non-zero. An example 3×3 polynomial matrix, degree 2:

$$P = \begin{pmatrix} 1 & x^2 & x \\ 0 & 2x & 2 \\ 3x+2 & x^2-1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 2 & -1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 3 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} x^2.$$

We can express this by saying that for a ring R, the rings $M_n(R[X])$ and $(M_n(R))[X]$ are isomorphic.

Properties [edit]

- A polynomial matrix over a field with determinant equal to a non-zero element of that field is called unimodular, and has an inverse that is also a polynomial matrix. Note that the only scalar unimodular polynomials are polynomials of degree 0 – nonzero constants, because an inverse of an arbitrary polynomial of higher degree is a rational function.
- The roots of a polynomial matrix over the complex numbers are the points in the complex plane where the matrix loses rank.

Note that polynomial matrices are *not* to be confused with monomial matrices, which are simply matrices with exactly one non-zero entry in each row and column.

If by λ we denote any element of the field over which we constructed the matrix, by I the identity matrix, and we let A be a polynomial matrix, then the matrix $\lambda I - A$ is the **characteristic matrix** of the matrix A. Its determinant, $|\lambda I - A|$ is the characteristic polynomial of the matrix A.

References [edit]

- E.V.Krishnamurthy, Error-free Polynomial Matrix computations, Springer Verlag, New York, 1985

This linear algebra-related article is a stub. You can help Wikipedia by expanding it.

Categories: Matrices | Polynomials | Linear algebra stubs

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
日本語
Русский
Slovenščina
Suomi
தமிழ்
ไทย
中文
Edit links

Given $p_1(x)$ with degree d_1,
$p_2(x)$ with degree d_2

$d = d_1 - d_2 \leq \max(d_1, d_2)$

Take a set S of random numbers,
Assuming $|S| = 100d$ (size of S)

if $p(r) = 0$ then output $p = 0$
if $p(r) \neq 0$ then output $p \neq 0$

Errors occur if $p \neq 0$ but $p(r) = 0$

1. Evaluate $p(x)$ in $O(n^2)$ trivially
$O(n)$ using Newton's Method

2. Euclidean alg for poly
$\deg(p(x)) = n$ $\deg(p'(x)) \leq n$ $p' \equiv 0$

Longdivision
$p(x) = p'(x) g(x) + r(x)$ $\deg(r(x)) < \deg(p'(x))$

$g(x)$, $r(x)$ unique

a is root of $p \iff p(a) = 0$

$\Rightarrow p(x) = (x - a) q(x)$ where $\deg(q(x)) \le n-1$

$p(x)$ 有 $\le n$ 个不同根

Problem: Given p_1, p_2, p_3 in var x

$n \quad n$

verify if $p_1(x) p_2(x) \equiv p_3(x)$

$\deg(p_3(x)) \le 2n$

1. Choose random num from a set S of size $\ge 4n+1$
2. Compute $p_1(r), p_2(r), p_3(r)$ $\quad r$
3. Output if $p_1(r) p_2(r) = p_3(r)$

Error may occur if true answer is not equal, but $p_1(r) p_2(r) = p_3(r)$ (r is root of

$p_1(r)p_2(r) = p_3(r)$)

Claim $\text{Prob}(\text{error}) \le \frac{1}{2}$

proof $p_1(x)p_2(x) - p_3(x)$ may not $\equiv 0$,
but it has $\le 2n$ roots

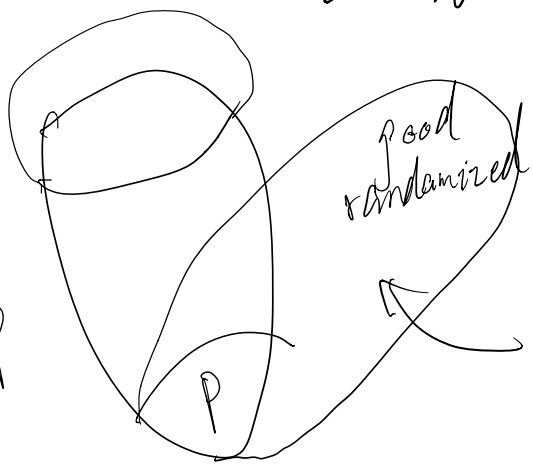
prob(picking r in step I which is a root of $p_1p_2 - p_3$)

$$\le \frac{2n}{4n+1} < \frac{1}{2}$$

We can increase the size of S.

PIT give you a good randomized algorithm
but there is no known P Time alg.

PIT is not known in NP.



PIT might be here.

Multivariate

$$p(x_1, x_2, \cdots x_n) = \sum \left(\prod_{i=1}^{n} c_i x_i^t \right)$$

$$p(x_1, x_2 \cdots x_n) = 3x_1^2 x_4^{14} + 2x_1 x_2 x_2^4 - \frac{14}{3} x^2 x^4$$

degree of a term $\frac{1}{10} x_1^5 x_2^2 x_3^1 x_4^2$

$5 + 2 + 1 + 2 = 10$

deg of poly is the degree of largest degree term with coeffi $\neq 0$