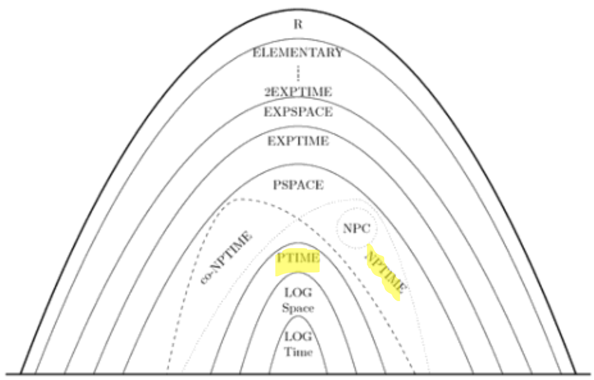
Lecture 1

Wednesday, 3 May 2023 11:16

Algebraic Methods in Computational Complexity Theory

— Abhiram Natarajan

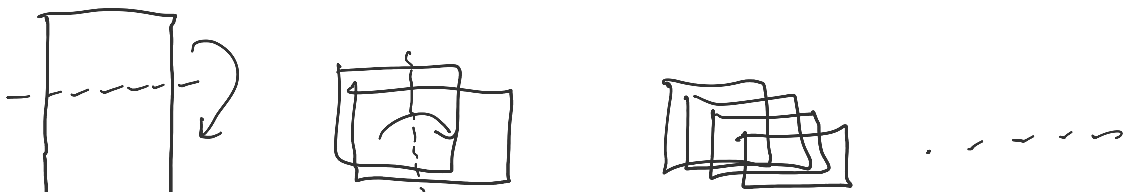


"Mathematical study of Computation"

TED TALK COMPLEXITY THEORY

$$1+2+\dots = -1/12$$

Process A



Step 1 Step 2 step 3

Process B

step 1
10000 sheets of paper

step 2
add 20000 sheets of paper on top

step 3
add 30000 sheets of paper

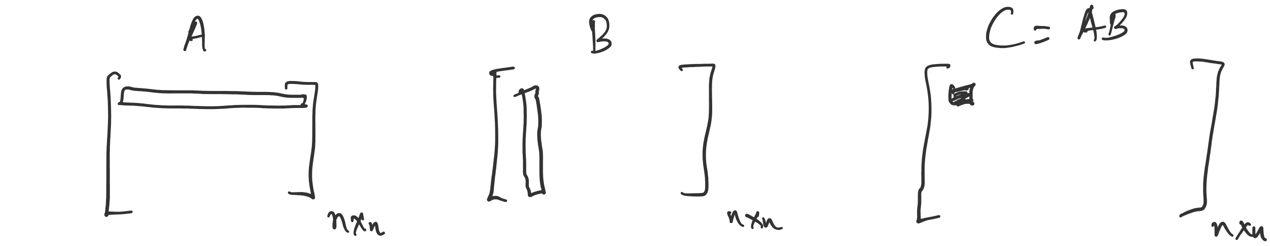
. . . .

* Process A takes 41 steps to get from earth to the moon
* Process B " 17 steps ——— " ———

* Process A takes 50 steps to get to the sun
* Process B " 387 " " " " " "

COOL CONTRIBUTIONS OF COMPLEXITY TO SCIENCE

(1) Matrix Multiplication



each entry requires n multiplications.
n^2 entries in C, thus n^3 operation are enough to compute the product.

Timeline of matrix multiplication exponent

Year	Bound on omega	Authors
1969	2.8074	Strassen[1]
1978	2.796	Pan[11]
1979	2.780	Bini, Capovani [it], Romani[12]
1981	2.522	Schönhage[13]
1981	2.517	Romani[14]
1981	2.496	Coppersmith, Winograd[15]
1986	2.479	Strassen[16]
1990	2.3755	Coppersmith, Winograd[17]
2010	2.3737	Stothers[18]
2013	2.3729	Williams[19][20]
2014	2.3728639	Le Gall[21]
2020	2.3728596	Alman, Williams[3]
2022	2.37188	Duan, Wu, Zhou[2]

2. CT gave us P vs NP question

Classic Nintendo Games are (NP-)Hard

Greg Aloupis* Erik D. Demaine† Alan Guo†‡

March 9, 2012

Abstract

We prove NP-hardness results for five of Nintendo's largest video game franchises: Mario, Donkey Kong, Legend of Zelda, Metroid, and Pokémon. Our results apply to Super Mario Bros. 1, 3, Lost Levels, and Super Mario World; Donkey Kong Country 1–3; all Legend of Zelda games except Zelda II: The Adventure of Link; all Metroid games; and all Pokémon role-playing games. For Mario and Donkey Kong, we show NP-completeness. In addition, we observe that several games in the Zelda series are PSPACE-complete.

expert in Donkey Kong / Super Mario Bros $\Longrightarrow$ Solve the Riemann Hyp.

3. Zero Knowledge proofs.

3. Zero Knowledge proofs.

You can convince someone that you have a proof of any mathematical theorem without revealing anything abt. the proof.

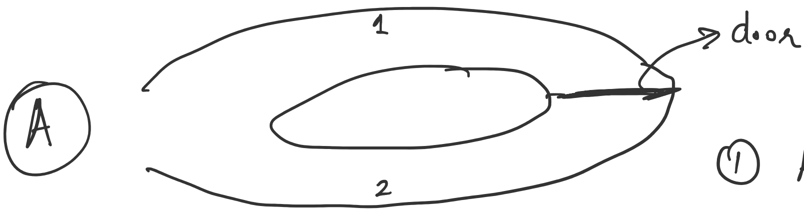
A YOU

Goal :- Convince skeptic that balls are differently coloured

1. Ball 1 in L Ball 2 in R, A shows you
2. A hides balls behind back, & either swaps the balls or not.
3. A shows hands again. You call out SWAP / NO SWAP
4. Repeat

A :- is Convinced if you answer correctly all times.

NO INFORMATION LEAKAGE



You know the pwd to door

1. A lets you get inside Cave
2. A asks you to come along 1/2.
3. Repeat.

A is Convinced if all calls are successfully answered.

4. Probabilistically Checkable proofs

You can write down the proof of the R.H. in such a way that anyone can verify the proof by looking at some like 60 bits of the proofs.

Prob. of being falsely convinced $\leq$ prob. that hallucinating

"PCP THEOREM"

5. P vs NP question

"IS DOING HARDER THAN CHECKING"

$\equiv$ Seperating orbit closures by calculating multiplicities of irreps

GCT program 'String theory of CS'

Methods that don't work:-

1. Relativization
2. Natural proofs barrier
3. Algebrization.

WORK AROUND P vs NP has been unique

OVERVIEW OF TOPICS I WANT TO COVER

1. Matrix Multiplication

$$A_{n\times n} * B_{n\times n} = C_{n\times n}$$

Naive method $\quad c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}$

n mult, $n-1$ additions per dot prod.

$\Rightarrow O(n^3)$ operations

Thm [Klyuyev, ~~Kokorkin~~ Schreebak '65] Naive method is optimal if only allowed to operate on rows & columns as a whole

Thm [Strassen '69] Can do better! $O(n^{2.81})$ operations

Observation 2x2 matrices. Naive method 8 products & 4 sums.

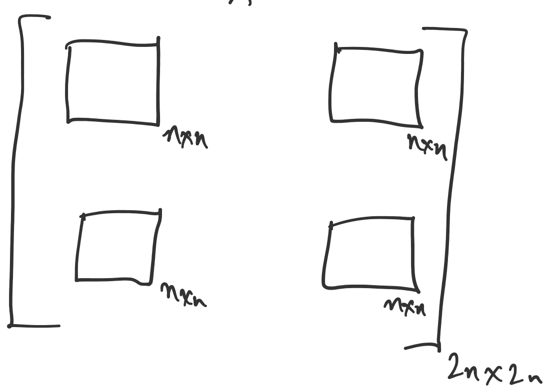
$$\begin{bmatrix} c_{1,1} \\ c_{1,2} \\ c_{2,1} \\ c_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} (a_{1,1}+a_{2,2})(b_{1,1}+b_{2,2}) \\ (a_{2,1}+a_{2,2})(b_{1,1}) \\ (a_{1,1})(b_{1,2}-b_{2,2}) \\ (a_{2,2})(-b_{1,1}+b_{2,1}) \end{bmatrix}$$

$$\begin{bmatrix} C_{2,1} \\ C_{2,2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} (a_{1,1})(b_{1,2}-b_{2,2}) \\ (a_{2,2})(-b_{1,1}+b_{2,1}) \\ (a_{1,1}+a_{1,2})(b_{2,2}) \\ (-a_{1,1}+a_{2,1})(b_{1,1}+b_{1,2}) \\ (a_{1,2}-a_{2,2})(b_{2,1}+b_{2,2}) \end{bmatrix}$$

Takes 7 products and 18 sums

Observation Above method works if $a_{i,j} \,|\, b_{i,j}$ are from non-comm. rings.

A B



$$M_2(M_n(\mathbb{C})) \cong M_{2n}(\mathbb{C})$$

- Thus you can recurse using Strassen's 2x2 idea, you can do matrix mult. using $O(n^{\log_2 7})$ operation

Defn 'ω' - called the exponent of $n\times n$ matrix mult.

$$\omega = \inf\{h \in \mathbb{R} \mid n\times n \text{ matrices can be multiplied using } O(n^h) \text{ operations}\}$$

$$2 \le \omega \le \underbrace{\log_2 7}_{\text{Strassen}}$$

Conjecture $\omega = 2$!!!

Abstraction of Matrix mult :-

Matrix mult. map $M_{\langle n\rangle} : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} \longrightarrow \mathbb{C}^{n\times n}$

— $M_{\langle n \rangle}$ is a bilinear map, thus it can be thought of as an elem of

$$(\mathbb{C}^{n\times n})^* \otimes (\mathbb{C}^{n\times n})^* \otimes (\mathbb{C}^{n\times n})$$

Thm ("geometric characterization of ω")

$$\omega = \inf \left\{ \tau \in \mathbb{R} \mid \text{rank } M_{\langle n \rangle} = O(n^\tau) \right\}$$

— Upper bounds correspond to decompositions of $M_{\langle n \rangle}$

— Lower bounds - study secant varieties of the Segre variety

(2) Defn ["efficient solving"] An algorithm for a problem X is called efficient if the no. of operations of the alg. is bounded from above by a polynomial in the size of the instance of problem X.

X is efficiently solvable if there exists an eff. alg. to solve X.

e.g. (1) Problem Take in a list of n numbers. Output any of them

Alg (1) Input the list → n operations

(2) Print the first one → 1 operation

$\sim n+1$ operations ✓✓ $\boxed{\text{efficient}}$

(2) Problem given a list of n number, output the list in descending order

Alg A (1) Generate all permutation of the list → $\geq n!$

(2) Check each permutation and o/p the one that is sorted $\geq n!$

$\sim \geq n!$ ✗✗ $\boxed{\text{inefficient}}$

Alg B (1) Find max of $a_1, \ldots a_n$, and output — n ops.

(2) Delete the max from the list, and repeat ...

$n-1$ ops

$n-2$ ops

$\vdots$

$O(n^2)$ ✓✓ $\boxed{\text{efficient}}$

③ Problem Check if n is prime $\log(n)$ - length of input

Alg: check for divisibility with all nos. from 2 to $\sqrt{n}$ — $\sqrt{n}$

total $\sqrt{n}$ XX $\boxed{\text{ineff}}$

$\downarrow$

exponential in $\log(n)$

There exists a poly alg. for primality.

Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin (2004). "PRIMES is in P" (PDF). *Annals of Mathematics*. **160** (2): 781–793.

Defn [Poly time reducibility]

If we can solve arbitrary instances of problem Y using a polynomial no. of steps, plus a polynomial no. of calls to an alg. that solves X, then we write

$$Y \leq_p X$$

"Y is polynomial time reducible to X" or "X is at least as hard as Y"

Fact if X is efficiently solvable, & $Y \leq_p X$, the Y is efficiently solvable too.

Defn [K-Satisfiability problem (K-SAT)]

Given $x_1, \dots x_n \in \{0,1\}$ variables

$C_1 \dots C_m$ clauses $m = O(n^c)$

$C_i = \bigvee_{j=1}^{K} t_{i,j}$ where $t_{i,j} \in \{x_1 \dots x_n, \bar{x}_1 \dots \bar{x}_n\}$

Ques Is there a satisfying assignment? $\{0,1\}^n$ s.t. all C_i are simultaneously satisfied

e.g. $x_1\ x_2\ x_3$

$C_1 = x_1 \vee \bar{x}_2$ $C_2 = \bar{x}_1 \vee \bar{x}_2$ $C_3 = x_2 \vee \bar{x}_3$

e.g. $x_1\ x_2\ x_3$

$$C_1 = x_1 \vee \bar{x}_2,\quad C_2 = \bar{x}_1 \vee \bar{x}_3,\quad C_3 = x_2 \vee \bar{x}_3$$

$(0,0,0)$ is a satisfying ass.

try all 2^n assignments NOT EFFICIENT

2-SAT is efficiently solvable

2. ^ a b c d e f Krom, Melven R. (1967), "The Decision Problem for a Class of First-Order Formulas in Which all Disjunctions are Binary", *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **13** (1–2): 15–20, doi:10.1002/malq.19670130104.

(≥ 3)-SAT is not known to be efficiently solvable.

Defn "Efficient Certification" — A soln. can be efficiently certified

$\Rightarrow$ the proposed soln can be checked in poly time.

We'll say X is efficiently certifiable if solutions to arbitrary problem instances are efficiently verified.

e.g. Factoring

$\rightarrow$ Solving efficiently unknown

$\rightarrow$ Certification efficient

Unsolved problem in computer science:

? *Can integer factorization be solved in polynomial time on a classical computer?*

(more unsolved problems in computer science)

3-SAT

$\rightarrow$ Solving efficiently unknown

$\rightarrow$ Certification is efficient

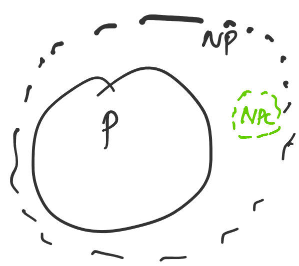
Def $P \rightarrow$ class of all problems efficiently solvable

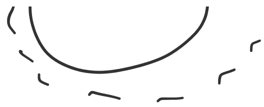
$NP \rightarrow$ class of all problems efficiently certifiable

$P \subseteq NP$, is converse true?

Conjecture $P \neq NP$

$P \subsetneq NP$





Defn X is an NP-Complete problem if

(1) $X \in NP$

(2) $\forall Y \in NP \quad Y \leq_p X$.

X is the hardest problems in NP.

Thm [Cook-Levin] Circuit-SAT is NP-Complete

Cor. Circuit-SAT $\leq_p$ 3-SAT

$\Rightarrow$ 3-SAT is NP-Complete

(*) If you can give an eff. alg for 3SAT $\Rightarrow P = NP$

(*) If you can show no such alg. exists $\Rightarrow P \neq NP$

(*) even showing 3-SAT requires $\omega(n)$ time is not doable at this pt.

(*) Not believed that you can do even slightly better than 2^n.

Exponential time hypothesis

5 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

In computational complexity theory, the **exponential time hypothesis** is an unproven computational hardness assumption that was formulated by Impagliazzo & Paturi (1999). It states that satisfiability of 3-CNF Boolean formulas cannot be solved in subexponential time, i.e., $2^{\epsilon n}$ for all constant $\epsilon > 0$,

WE SHALL WORK- ON A "MORE ALGEBRAIC" CONJECTURE

"algebraic analog of P vs NP"

Def VP – Class of polynomials that are easy to evaluate

VNP – class of polynomials whose co-effs are easy to evaluate

e.g $\det_n \in VP$ $\quad n!$ terms, Can be computed using Gaussian elim in $O(n^3)$

$$\mathrm{Perm}_n = \sum_{\sigma \in S_n} x_{1,\sigma(1)} x_{2,\sigma(2)} \cdots x_{n,\sigma(n)} \in VNP$$

Question [Valiant 1979]

$$VP \overset{?}{\subsetneq} VNP$$

For the relation b/w this & P vs NP, see Chap 21 of

$VP \neq \ldots$

For the relation b/w this & P vs NP, see Chap 21 of

Bürgisser, Peter; Clausen, Michael; Shokrollahi, M. Amin (1997). *Algebraic complexity theory*. Grundlehren der Mathematischen Wissenschaften. Vol. 315. With the collaboration of Thomas Lickteig. Berlin: Springer-Verlag. ISBN 978-3-540-60582-9. Zbl 1087.68568.

Thm [Valiant 1979]

$$VP \neq VNP \iff \mathrm{Perm}_n \notin VP$$

Notice $\det_n$ has $n!$ monomials, is poly(n) time computable

$\det_{n^c}$ has $n^c!$ monomials, is poly(n) time computable

Thus if Perm_n can be expressed as the determinant of a matrix of size poly in n, then $\mathrm{Perm}_n \in VP \Rightarrow VP = VNP$

Defn [Determinantal Complexity] $f \in \mathbb{F}[\bar{x}]$

$dc(f)$ is the min 's' s.t there is an $s \times s$ matrix of affine linear forms $\alpha_{i,j} \in \mathbb{F}[\bar{x}]$ $1 \leq i,j \leq s$

Such that

$$f = \det \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,s} \\ \vdots & & \\ \alpha_{s,1} & \cdots & \alpha_{s,s} \end{bmatrix}$$

We will show

$$\frac{n^2}{2} \leq dc(\mathrm{perm}_n) \leq 2^n - 1$$

Conj $dc(\mathrm{perm}_n)$ grows faster than any polynomial in n.

FEWNOMIALS

e.g. $f = 7x^{100} - 22x^{32} + 45x^{21} + 9$

Thm [Descartes Rule of Signs]

$f \in \mathbb{R}[x]$, no of real roots (counted with multiplicity)

$\sim 2t$

$\downarrow$

sparsity .

Thm [MULTIVARIATE GENERIZATIONS]

"Bezout theorem for fewnomials"

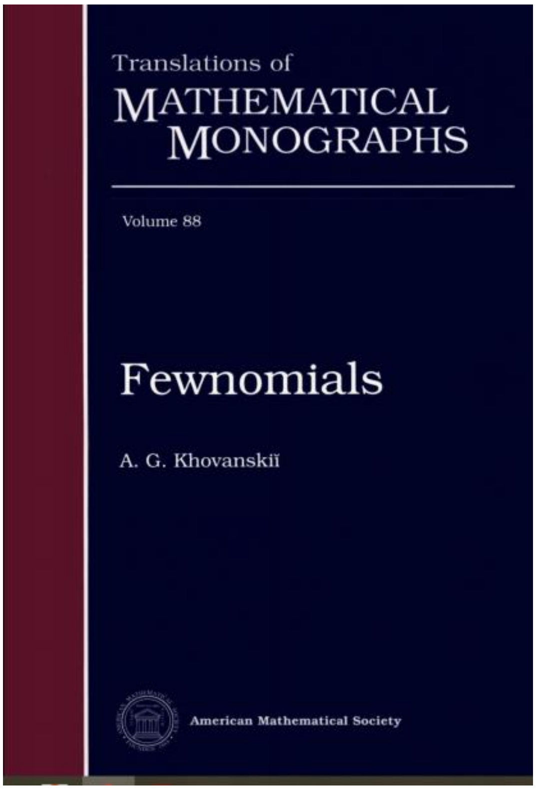
$f_1 \dots f_n \in \mathbb{R}[x_1 \dots x_n]$

$\left\{ f_i = 0 \right.$ has at most $l+n+1$ distinct monomials

System has at most

$$2^{\binom{l+n}{2}} (n+1)^{l+n}$$ no. of

non-singular solutions in the positive orthant of $\mathbb{R}^n$



Conjecture [Real-Tau Conjecture - Koiran] Consider polynomials $f_{i,j} \in \mathbb{R}[x]$

that are t-sparse.

$$f = \sum_{i=1}^{m} \prod_{j=1}^{k} f_{i,j}$$

no. of real zeros of $f \leq \text{Poly}(m, t, 2^k)$

Thm [Koiran]

Conjecture $\Rightarrow VP_{\mathbb{C}} \neq VNP_{\mathbb{C}}$

GCT

$V = (\mathbb{C}^{m^2})^*$

$GL(V)$ – group of all automorphisms on V

acts on

$\text{Sym}^m(V)$ – degree 'm' homogeneous polynomials in m^2-variables

$$\underset{\substack{\uparrow \\ GL(V)}}{L} \cdot \underset{\substack{\uparrow \\ \text{Sym}^m(V)}}{p(x)} := P(L^T x)$$

(b/c of transpose $L_1 \cdot (L_2 \cdot P) = (L_1 L_2) \cdot P$)

$\det_m \in \underline{\text{Sym}^m(V)}$

Let $n \leq m$. Per$_n$ is a lower degree poly than $\det_m$ in fewer vars. Define Padded permanent

$$\text{Per}^*_{m,n} = \underline{x_{m,m}^{m-n}\, \text{Perm}_n} \in \text{Sym}^m(V)$$

Conjecture [Mulmuley-Sohoni]

If $m = 2^{n^{o(1)}}$, then for sufficiently large n,

$$\text{Per}^*_{m,n} \notin \overline{GL(V) \cdot \det_m}$$

Thm $dc(\text{perm}_n) \leq m \Rightarrow \text{Per}^*_{m,n} \in \overline{GL(V) \cdot \det_m}$

Thm $dc(\mathrm{perm}_n) \leq m \implies \mathrm{Per}_{m,n}^{*} \in \overline{GL(V)\cdot \det_m}$