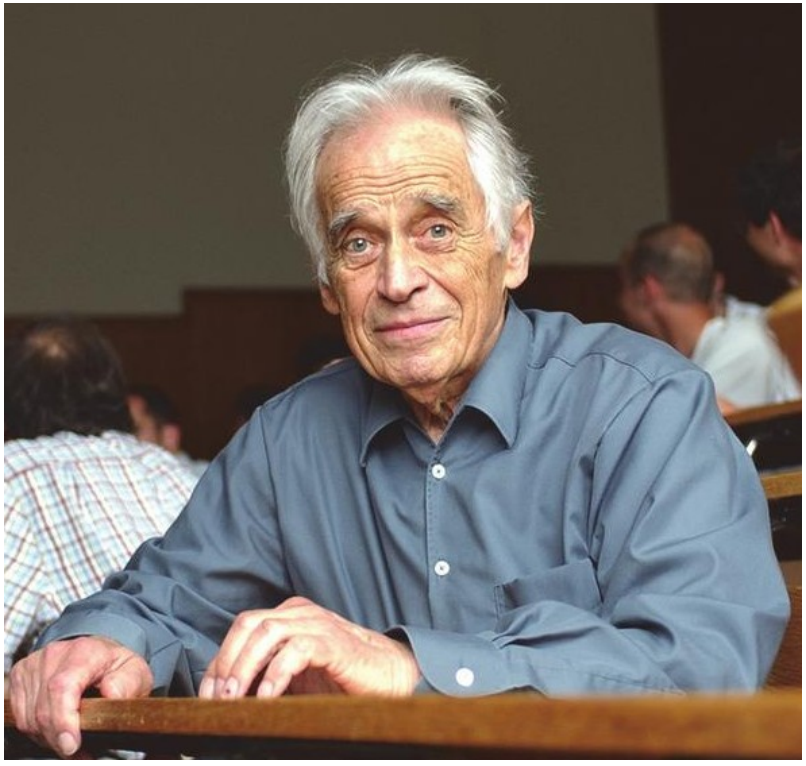
90 Years of Computability and Complexity

Stathis Zachos

National Technical University of Athens

190 years of Specker and Engeler together
February 21-22, 2020
Zürich

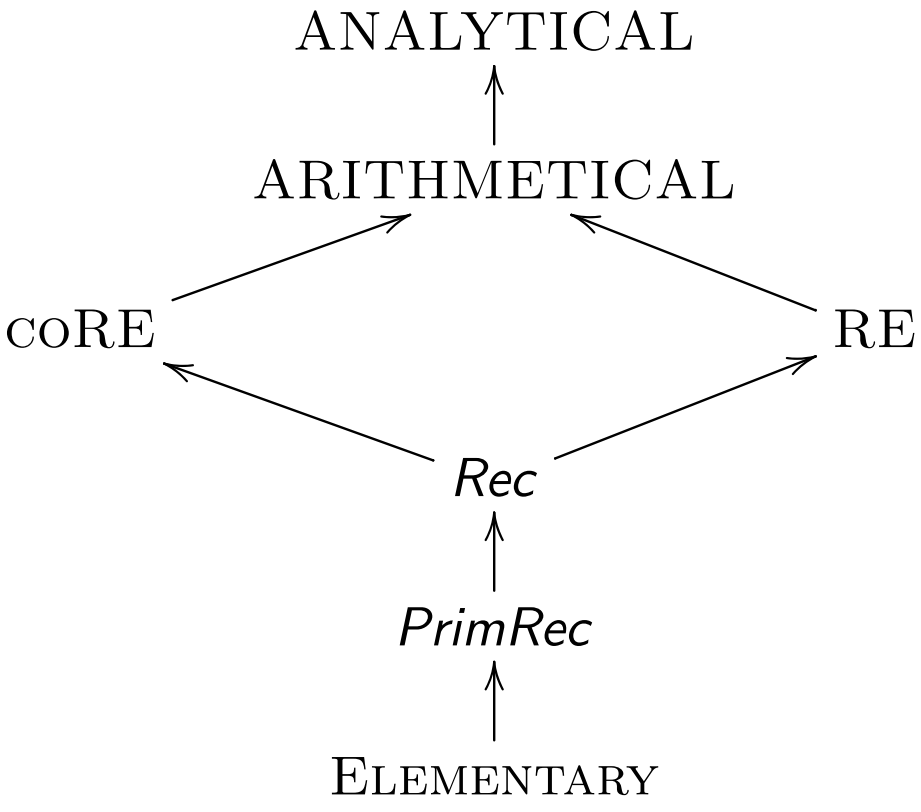


Abstract:

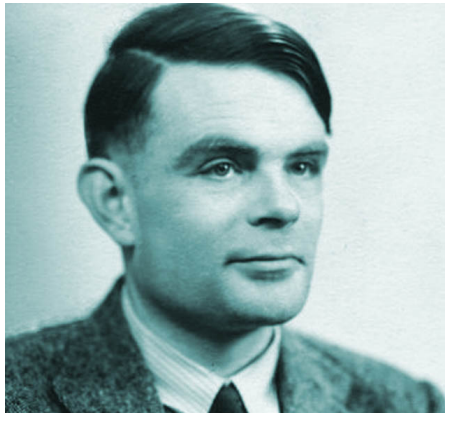
Computational Complexity Theory deals with the classification of problems into classes of hardness called complexity classes. We define complexity classes using general structural properties, such as the model of computation (Turing Machine, RAM, Finite Automaton, PDA, LBA, PRAM, monotone circuits), the mode of computation (deterministic, nondeterministic, probabilistic, alternating, uniform parallel, nonuniform circuits), the resources (time, space, # of processors, circuit size and depth) and also randomness, oracles, interactivity, counting, approximation, parameterization, etc. The cost of algorithms is measured by worst-case analysis, average-case analysis, best-case analysis, amortized analysis or smooth analysis.

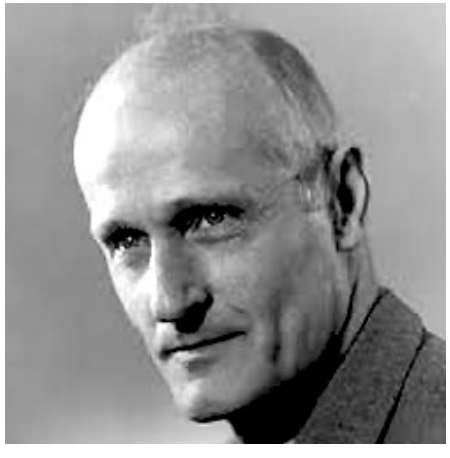
Inclusions and separations between complexity classes constitute central research goals and form some of the most important open questions in Theoretical Computer Science. Inclusions among some classes can be viewed as complexity hierarchies. We will present some of these: the Arithmetical Hierarchy, the Chomsky Hierarchy, the Polynomial-Time Hierarchy, a Counting Hierarchy, an Approximability Hierarchy and a Search Hierarchy.

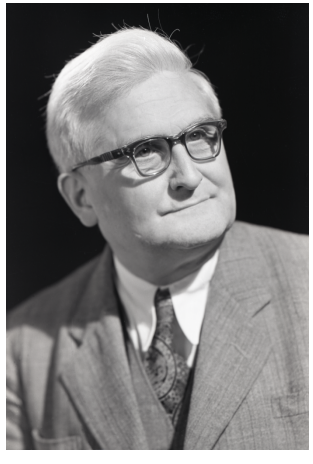
Gödel, Church, Kleene, Turing 30's, 40's: Unsolvability



Gödel, Church, Kleene, Turing, Kalmar 30's, 40's: Unsolvability







KALMAR ELEMENTARY:
Loop-Computable with number of nested for-loops ≤ 2

PrimRec: Primitive Recursive, Loop-Computable

Rec: Recursive, Decidable, Computable

RE: Recursively Enumerable, Listable, Acceptable

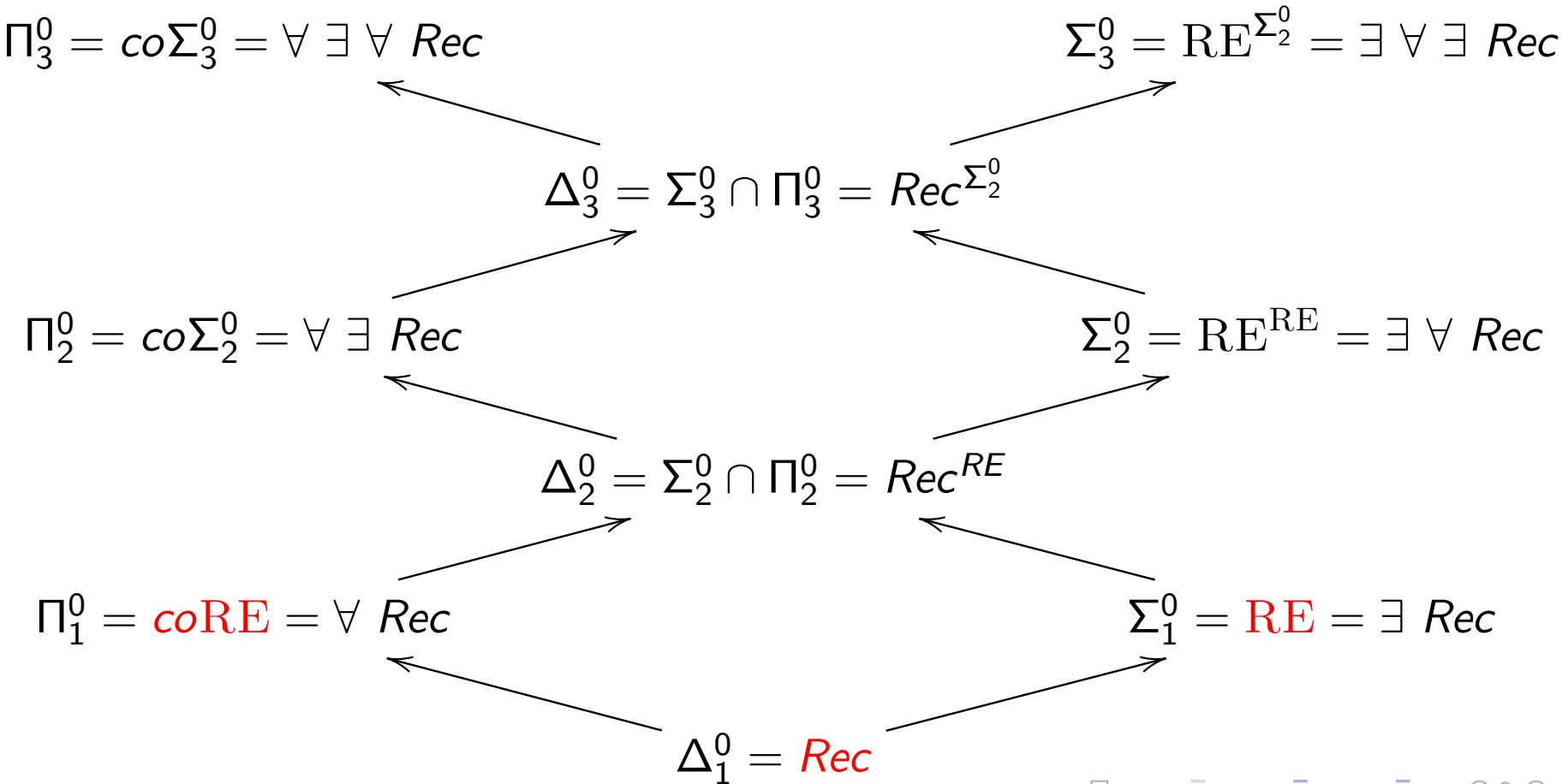
ARITHMETICAL:
Definable in Arithmetic: $\mathbb{N} = \langle N; <; S; +; *; 0 \rangle$.
Definable by first-order quantified formula over a recursive predicate. E.g.: $\exists x_1 \forall x_2 \exists x_3 \ldots R(x_1, \ldots, x_k) \in \Sigma_k^0$

ANALYTICAL: Definable by a second-order quantified formula.
E.g., $\exists$ set A, $\forall$ function f, ...

Kleene: Arithmetical Hierarchy

Oracle Notation vs. Quantifier Notation

$\ldots$



Chomsky, Rabin, Scott

50's: Formal Languages and Automata

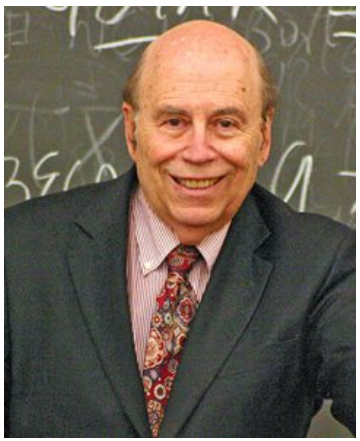
Deterministic vs. Nondeterministic Model

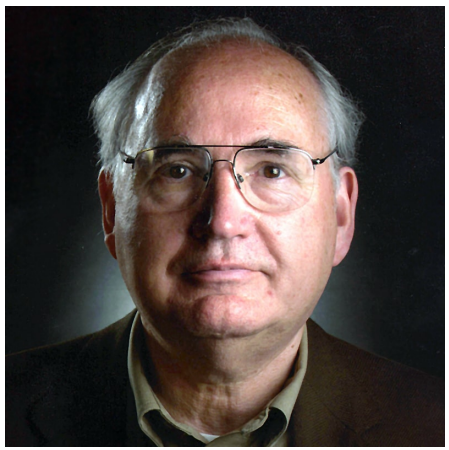
Relation of C with coC

$$
\begin{array}{ll}
\mathrm{RE} & \neq \mathit{co}\mathrm{RE} \\
\uparrow & \\
\mathrm{CS} & = \mathit{co}\mathrm{CS} \\
\uparrow & \\
\mathrm{CF} & \neq \mathit{co}\mathrm{CF} \\
\uparrow & \\
\mathrm{REG} & = \mathit{co}\mathrm{REG} \\
\uparrow & \\
\mathrm{FIN} & \neq \mathit{co}\mathrm{FIN}
\end{array}
$$

Chomsky, Rabin, Scott, Kleene

50's: Formal Languages and Automata





FIN: finite

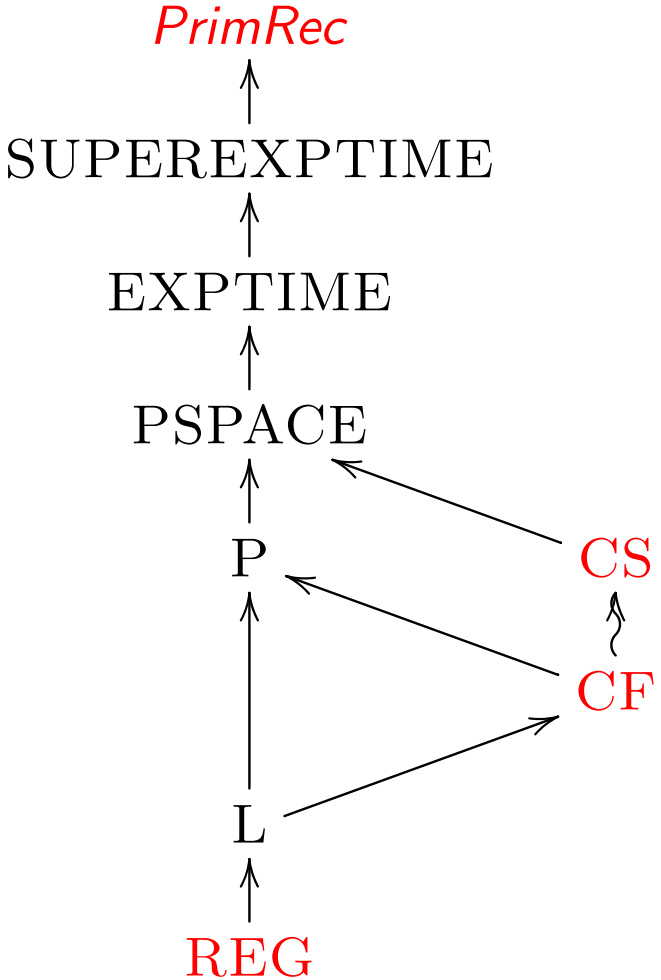
REG: decidable (acceptable) by a (Deterministic or Nondeterministic) Finite Automaton, equivalently definable by a Regular Expression, equivalently generatable by a Right-Linear Grammar

CF: decidable (acceptable) by a (Nondeterministic) Push-Down Automaton, equivalently generatable by a Context-Free Grammar

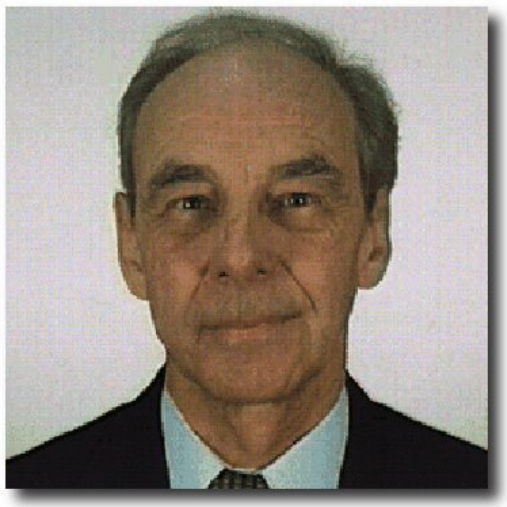
CS: decidable (acceptable) by a (Nondeterministic) Linearly-Bounded Automaton, equivalently generatable by a Context-Sensitive Grammar

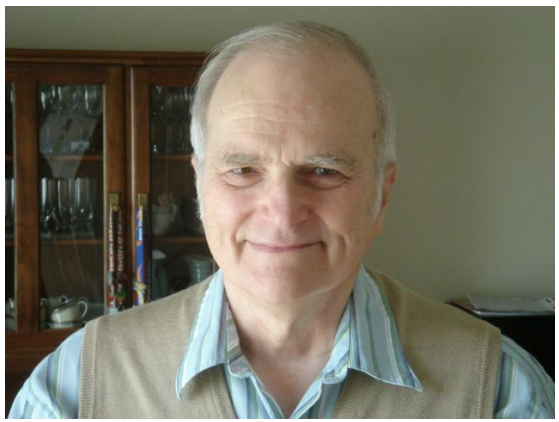
RE: acceptable by a (Deterministic or Nondeterministic) Turing Machine, equivalently generatable by a General Grammar

Hartmanis 60's: Computational Complexity (Space, Time)



Hartmanis 60's: Computational Complexity (Space, Time)





Hartmanis 60's: Computational Complexity (Space, Time)

$$\mathrm{SUPEREXPTIME} = \mathsf{DTIME}(2^{2^{\cdots^{2}}}\Big\}\, n \text{ times})$$

$$\mathrm{EXPTIME} = \bigcup_{i\geq 1} \mathsf{DTIME}(2^{n^i})$$

$$\mathrm{PSPACE} = \bigcup_{i\geq 1} \mathsf{DSPACE}(n^i)$$

$$\mathrm{P} = \bigcup_{i\geq 1} \mathsf{DTIME}(n^i)$$

$$\mathrm{L} = \mathsf{DSPACE}(\log n)$$

Hartmanis 60's: Computational Complexity (Space, Time)

Hierarchy Theorems (Deterministic and Nondet.)

Theorem (Hartmanis, Lewis, Stearns, 1965)

$SPACE(o(s(n))) \subsetneq SPACE(s(n))$

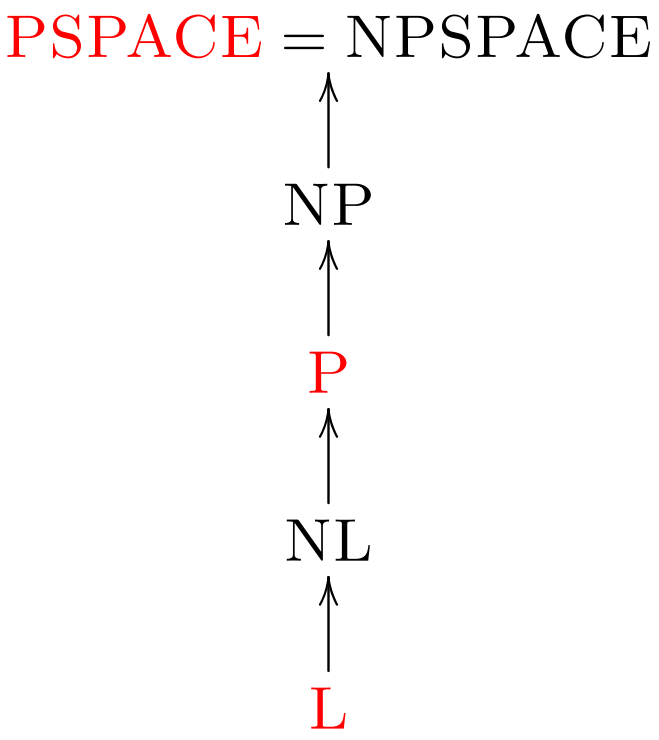
Theorem (Fürer, 1982)

$TIME(o(t(n))) \subsetneq TIME(t(n))$

Hartmanis 60's: Computational Complexity (Space, Time)

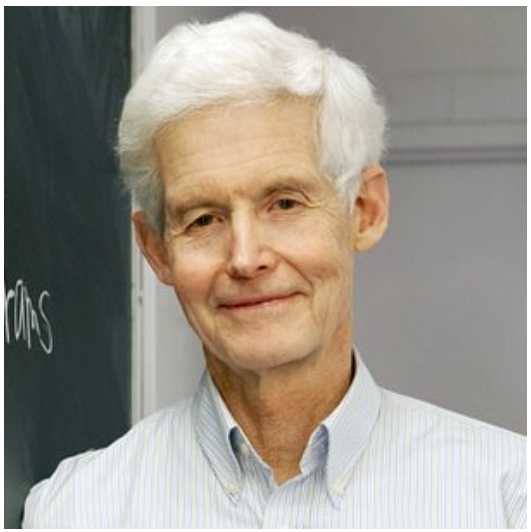
Cook, Karp, Savitch

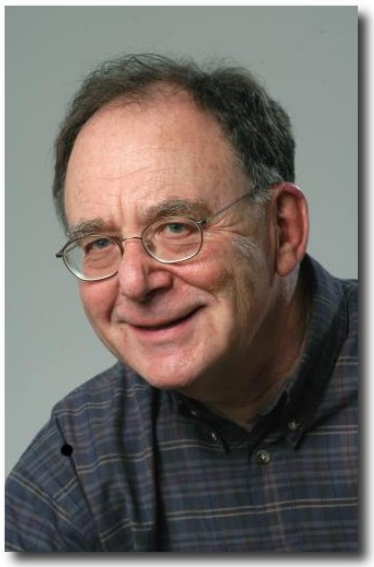
early 70's: Nondeterminism and Complexity, NP-completeness



Cook, Karp, Savitch

early 70's: Nondeterminism and Complexity, NP-completeness





$$\mathrm{NP} = \bigcup_{i \geq 1} \mathrm{NTIME}(n^i)$$

$$\mathrm{NL} = \mathrm{NSPACE}(\log n)$$

Oracles

$$\mathrm{P}^A$$

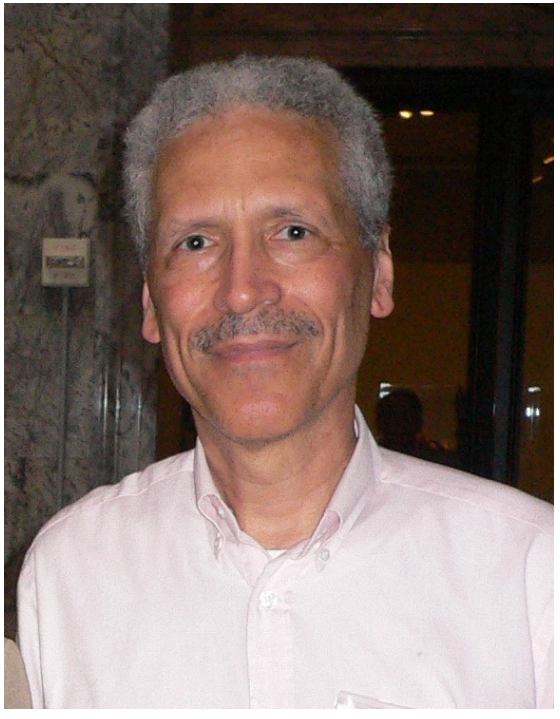
$$\mathrm{NP}^A$$

$$\mathrm{P}^{SAT} = \mathrm{P}^{\mathrm{NP}}$$

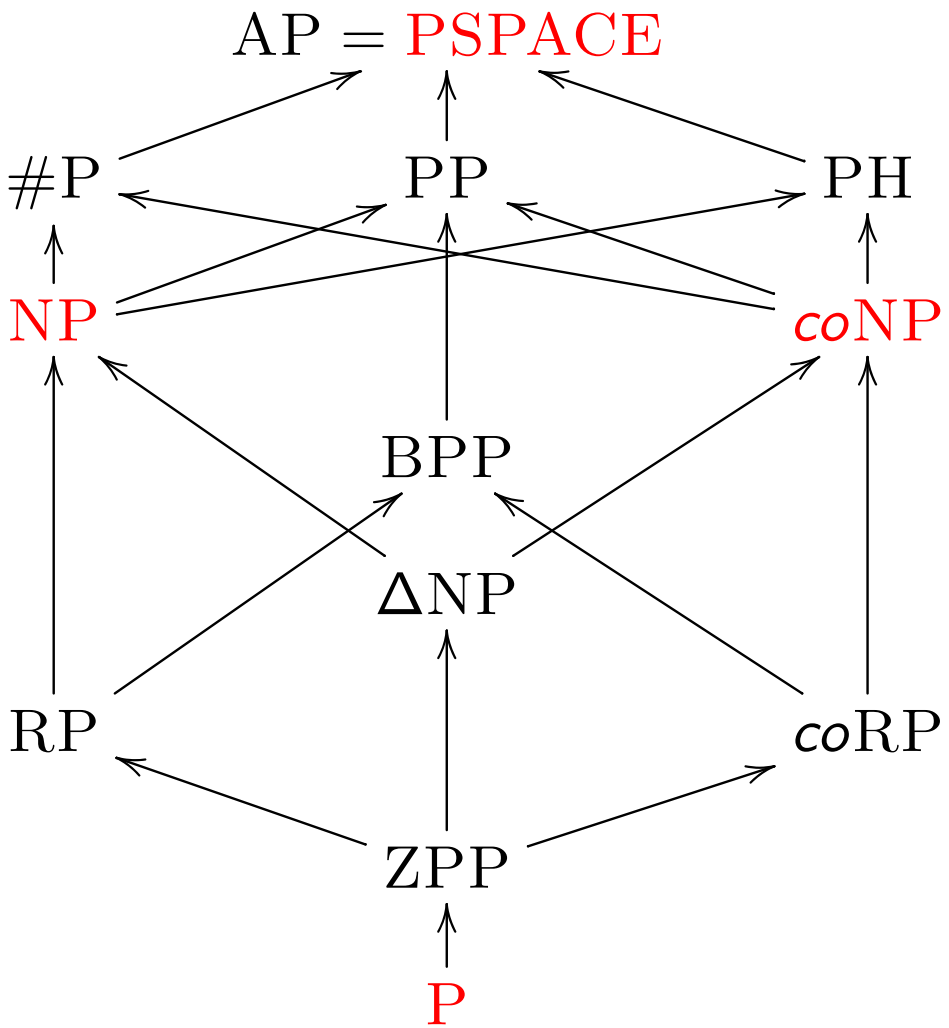
$$\mathrm{NP}^{SAT} = \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$$

Solovay, Gill

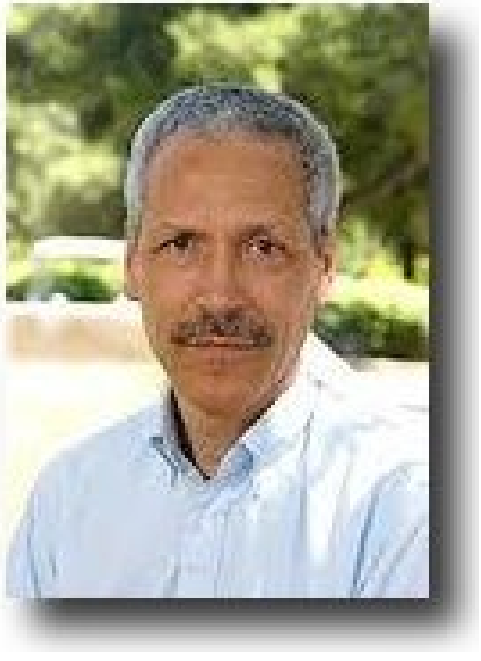
early 70's: Inclusions and Separations with Oracles



Stockmeyer, Valiant, Gill late 70's: Probabilistic, Polynomial Hierarchy, Counting, Alternation



Stockmeyer, Valiant, Gill late 70's: Probabilistic, Polynomial Hierarchy, Counting, Alternation



BPP: Bounded-error Probabilistic Polynomial (both-sided error possible), also known as Monte Carlo

$$L \in \mathrm{BPP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L \implies \exists^{+}\ y\ R(x,y) \\ x \notin L \implies \exists^{+}\ y\ \neg R(x,y) \end{cases}$$

RP: Randomized Polynomial (one-sided error)

$$L \in \mathrm{RP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L \implies \exists^{+}\ y\ R(x,y) \\ x \notin L \implies \forall\ y\ \neg R(x,y) \end{cases}$$

ZPP: Zero-error Probabilistic Polynomial, also known as Las Vegas

$\mathrm{ZPP} = \mathrm{RP} \cap co\mathrm{RP}$

$\Delta\mathrm{NP} = \mathrm{NP} \cap co\mathrm{NP}$, in general $\Delta\mathcal{C} = \mathcal{C} \cap co\mathcal{C}$

PP: Probabilistic Polynomial (the possibility of error is not bounded away from 1/2); not a practical class

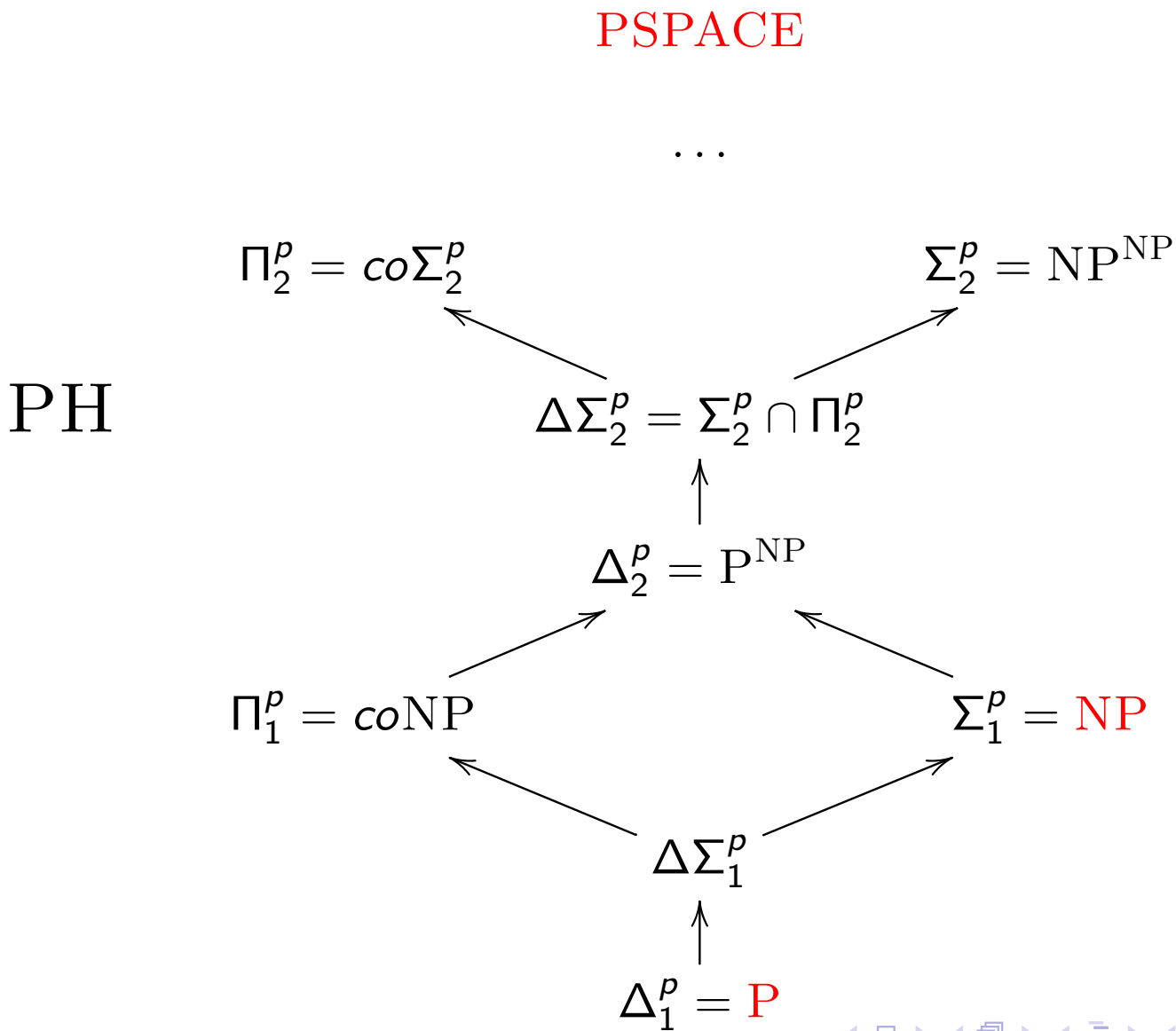
$$L \in \mathrm{PP} \iff \exists R \in \mathrm{P}: \begin{cases} x \in L \implies \exists_{1/2}\ y\ R(x,y) \\ x \notin L \implies \exists_{1/2}\ y\ \neg R(x,y) \end{cases}$$

PH: Polynomial Hierarchy

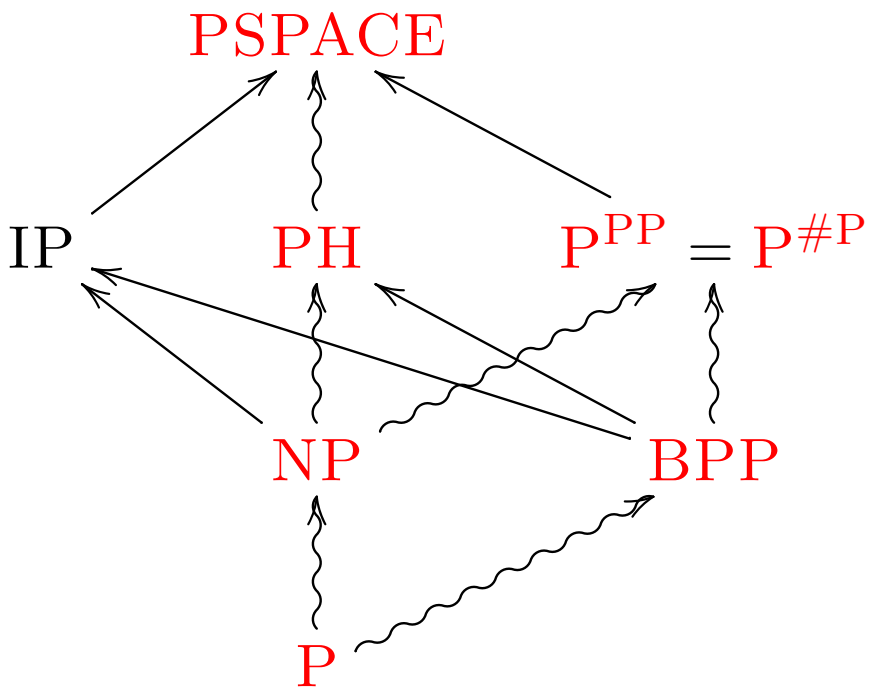
#P: the class of functions f for which there is a polynomial time NDTM, whose computation tree has exactly $f(x)$ accepting computation paths (for input x).

AP: Alternating (Turing Machine) Polynomial Time

Stockmeyer: Polynomial Hierarchy



Goldwasser, Micali, Rackoff, Sipser, Wigderson, Z.
early 80's: Interactive Proofs



Goldwasser, Micali, Rackoff, Sipser, Wigderson, Z.

early 80's: Interactive Proofs

$L \in \mathrm{IP}$:

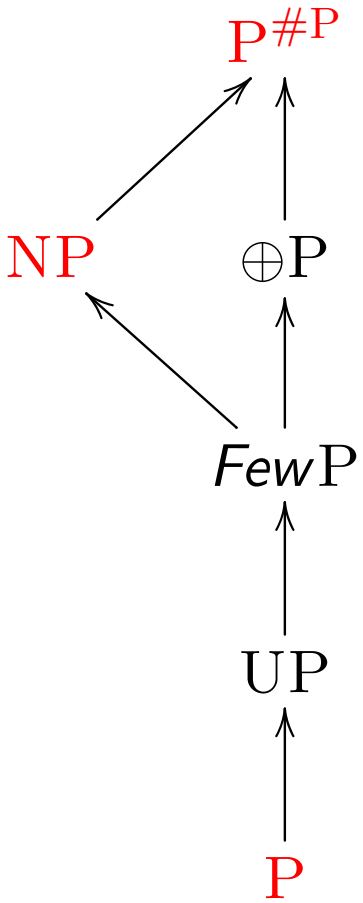
- $x \in L \implies \exists$ prover P, such that verifier V accepts with overwhelming probability.
- $x \notin L \implies \forall$ prover P, verifier V does not accept with overwhelming probability.

It has been shown that the first condition can be equivalently formulated:

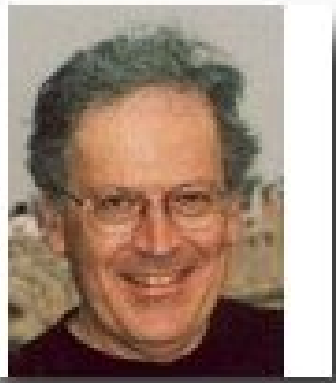
- $x \in L \implies \exists$ prover P, such that verifier V always accepts (i.e., with probability 1)

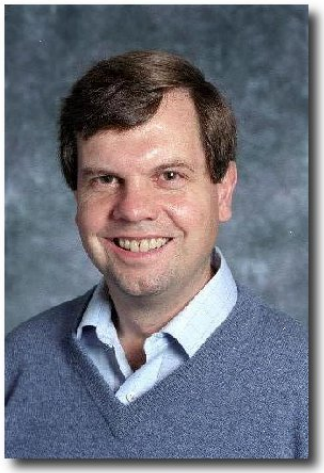
PP and $\#\mathrm{P}$ are Cook-interreducible

Valiant, Vazirani[2], Papadimitriou, Allender, Z.
80's: Counting classes, One-Way Functions



Valiant, Vazirani[2], Papadimitriou, Allender, Z.
80's: Counting classes, One-Way Functions





The model is a nondeterministic polynomial time TM.
The computation tree on input x s a full complete binary tree of height $p(|x|)$.

$\oplus\mathrm{P}$: if answer is 'yes' then # accepting paths is odd, if answer is 'no' then # accepting paths is even (parity)

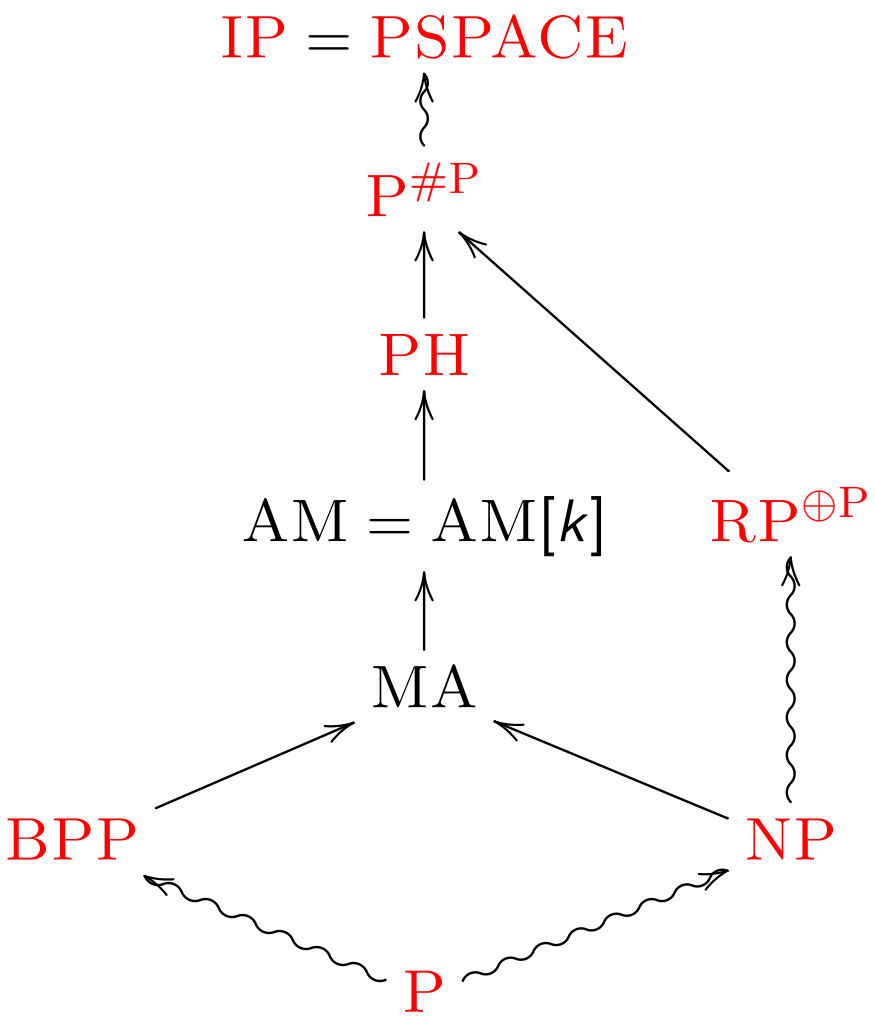
Few$\mathrm{P}$: if answer is 'yes' then # accepting paths is bounded by a polynomial w.r.t. size of input (fewness)

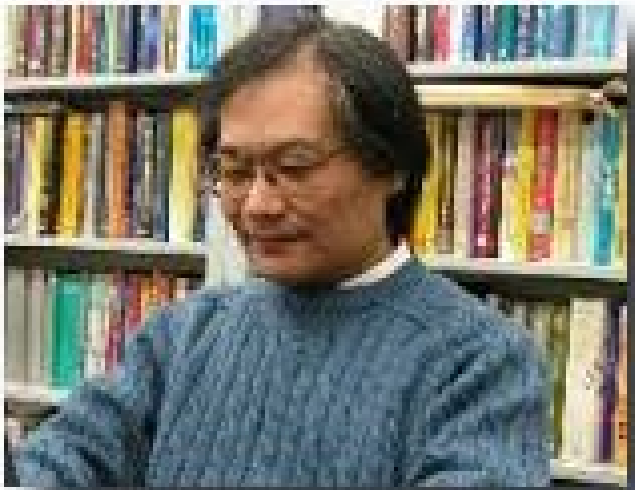
UP: if answer is 'yes' then exactly one accepting path (uniqueness)

Theorem (Valiant - V. Vazirani): $\mathrm{NP} \subseteq \mathrm{RP}^{\oplus\mathrm{P}}$

Babai, Toda, Shamir, Z.

80's, 90's: Arthur-Merlin, Classification of IP and PH





Merlin: Prover Arthur: Verifier

$L \in \mathrm{AM}(k)$ iff $\exists$ a k-move game where Arthur plays first and:

- $x \in L \implies$ Arthur is convinced with overwhelming probability that $x \in L$
- $x \notin L \implies$ With overwhelming probability Arthur is not convinced that $x \in L$.

It has been shown that the first condition can be equivalently formulated:

- $x \in L \implies$ Arthur is convinced with probability 1

$$\mathrm{AM} = \mathrm{AM}(2) \qquad \mathrm{MA}$$

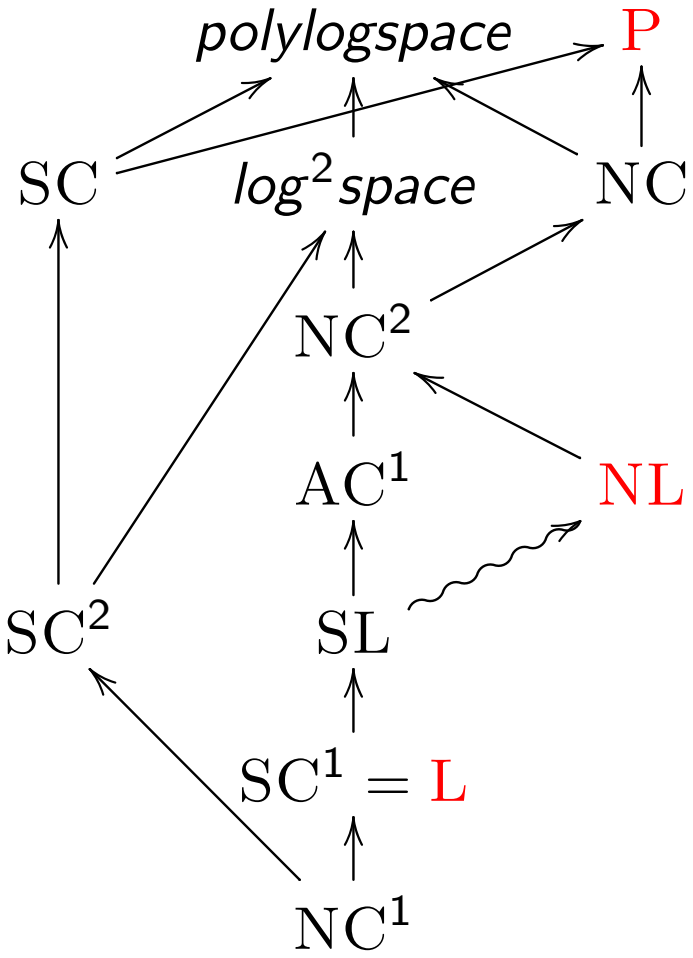
Immerman, Szelepcsényi

$$\mathrm{NSPACE}(S(n)) = co\mathrm{NSPACE}(S(n))$$

Corollary: $\mathrm{CS} = co\mathrm{CS}$ LBA problem

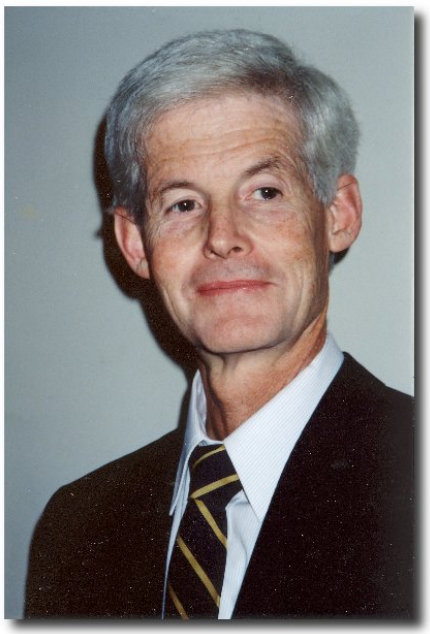
Pippenger, Cook, Borodin

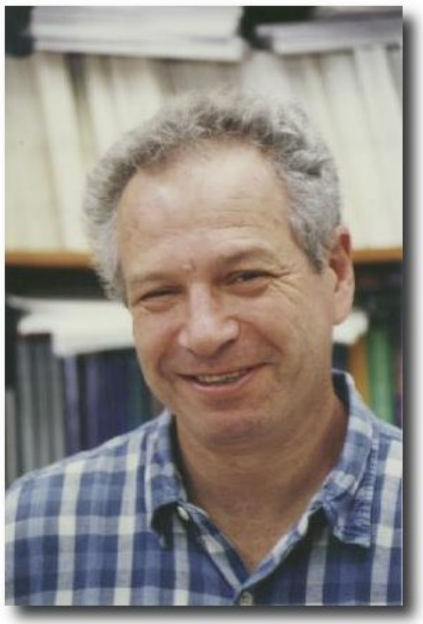
80's: Below P (uniform circuit families)



Pippenger, Cook, Borodin

80's: Below P (uniform circuit families)





($k \geq 0$):

1. NC^k: class of languages acceptable by $\mathrm{DLOGTIME}$-uniform circuit families of polynomial size and $\mathcal{O}(\log^k n)$ depth, using bounded fan-in gates.

2. AC^k: class of languages acceptable by $\mathrm{DLOGTIME}$-uniform circuit families of polynomial size and $\mathcal{O}(\log^k n)$ depth, using unbounded fan-in gates.

3. TC^k: class of languages acceptable by $\mathrm{DLOGTIME}$-uniform circuit families of polynomial size and $\mathcal{O}(\log^k n)$ depth, using threshold gates.

4. SC^k: class of languages acceptable by a DTM in polynomial time and in $\mathcal{O}(\log^k n)$ space.

$$\mathrm{NC} = \bigcup_{k \geq 0} \mathrm{NC}^k$$

$$\mathrm{AC} = \bigcup_{k \geq 0} \mathrm{AC}^k$$

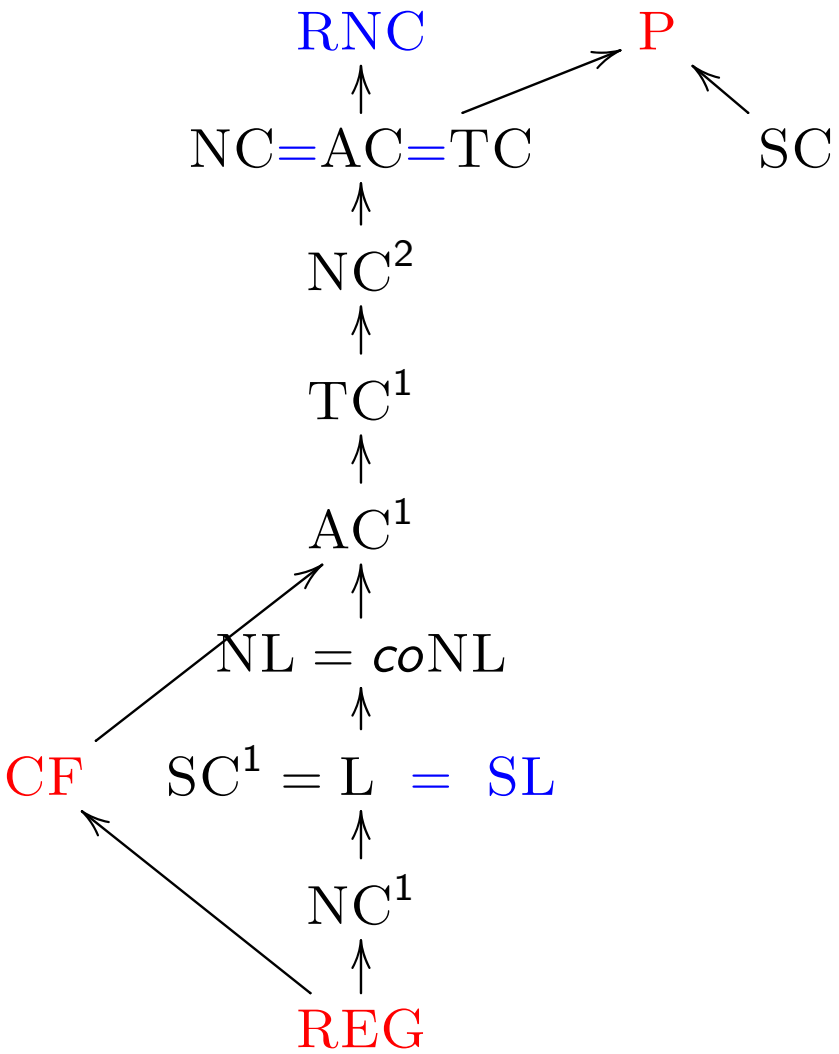
$$\mathrm{TC} = \bigcup_{k \geq 0} \mathrm{TC}^k$$

$$\mathrm{SC} = \bigcup_{k \geq 0} \mathrm{SC}^k$$

SL (Symmetric logspace):
all problems decidable by a symmetric logspace TM or
all problems reducible to undirected s-t connectivity

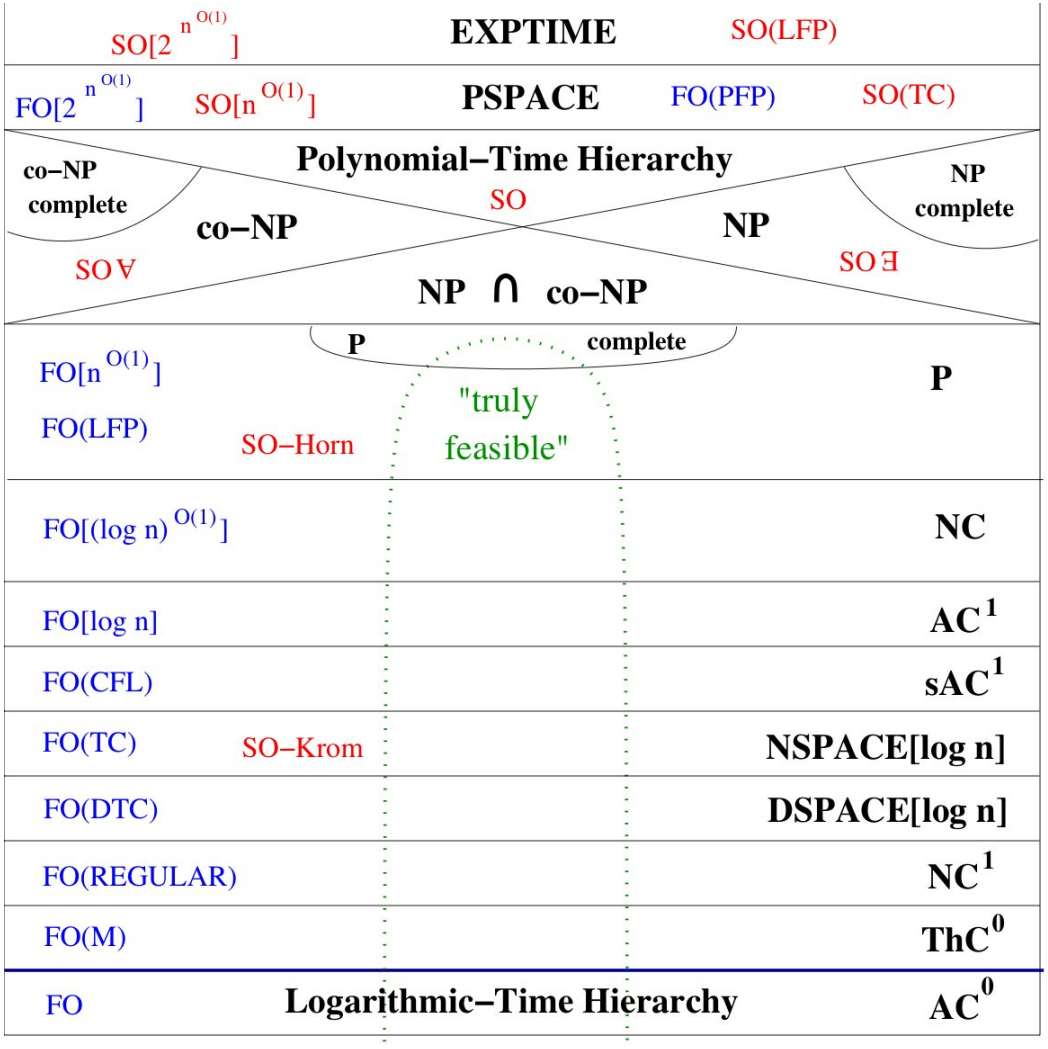
RNC (Randomized NC):
has the same relation to NC as RP has to P

Pippenger, Reingold (2004) 80's/90's: Connections



Pippenger, Reingold (2004) 80's/90's: Connections

Fagin, Immerman, Kolaitis, Vardi, Grädel
Expressibility and Descriptive Complexity



Fagin, Immerman, Kolaitis, Vardi, Grädel
Expressibility and Descriptive Complexity

Yannakakis, Papadimitriou, Arora, Sudan, Safra, Dinur
90's: PCP and Approximation

NPO
↑
*poly*APX
↑
*log*APX
↑
APX
↑
APTAS
↑
PTAS
↑
FPTAS
↑
PO

Yannakakis, Papadimitriou, Arora, Sudan, Safra, Dinur
90's: PCP and Approximation

NPO: the class of optimization problems for which the underlying decision problem is in NP (with the condition that there are feasible solutions for every instance)

PO: the class of optimization problems for which the underlying decision problems is in P

APX: problems for which there exists a ρ-approximative algorithm for some constant $\rho > 0$

log-APX: problems for which there exists a $\log n$-approximative algorithm (where n is the input size: $n = |x|$)

poly-APX: problems for which there exists a $p(n)$-approximative algorithm for some polynomial p (where n is the input size: $n = |x|$)

PTAS: problems for which there exists a *polynomial time approximation scheme*, i.e.,a $(1+\epsilon)$-approximative algorithm *for any* constant $\epsilon > 0$

FPTAS: problems for which there existsa *fully polynomial time approximation scheme*, i.e., a $(1+\epsilon)$-approximative algorithm *for any* constant $\epsilon > 0$, where, the time needed is also polynomial w.r.t. $1/\epsilon$

APTAS: problems for which there exists an *asymptotic polynomial time approximation scheme*, i.e., a $(1 + \epsilon + \frac{c}{OPT})$-approximative algorithm *for any* constant $\epsilon > 0$, for some constant c

$L \in \mathrm{PCP}$ (Probabilistically Checkable Proofs):

- $x \in L \implies \exists$ proof Π such that the verifier V always accepts (i.e., with probability 1),
- $x \notin L \implies \forall$ 'proof' Π, the verifier V does not accept with overwhelming probability.

$\mathrm{PCP}(r(n), q(n))$ consists of languages $L \in \mathrm{PCP}$ such that the polynomial time verifier V uses $O(r(n))$ random bits and queries $O(q(n))$ bits of the proof.

$\mathrm{PCP} = \mathrm{PCP}(\mathrm{poly}(n), \mathrm{poly}(n)) = \mathrm{MIP} = \mathrm{NEXP}$

$\mathrm{P} = \mathrm{PCP}(0, 0)$ $\qquad$ $\mathrm{NP} = \mathrm{PCP}(0, \mathrm{poly}(n))$

$co\mathrm{RP} = \mathrm{PCP}(\mathrm{poly}(n), 0)$

Theorem (Arora, Lund, Motwani, Sudan, Szegedy)

$$\mathrm{NP} = \mathrm{PCP}(\log n, 1)$$

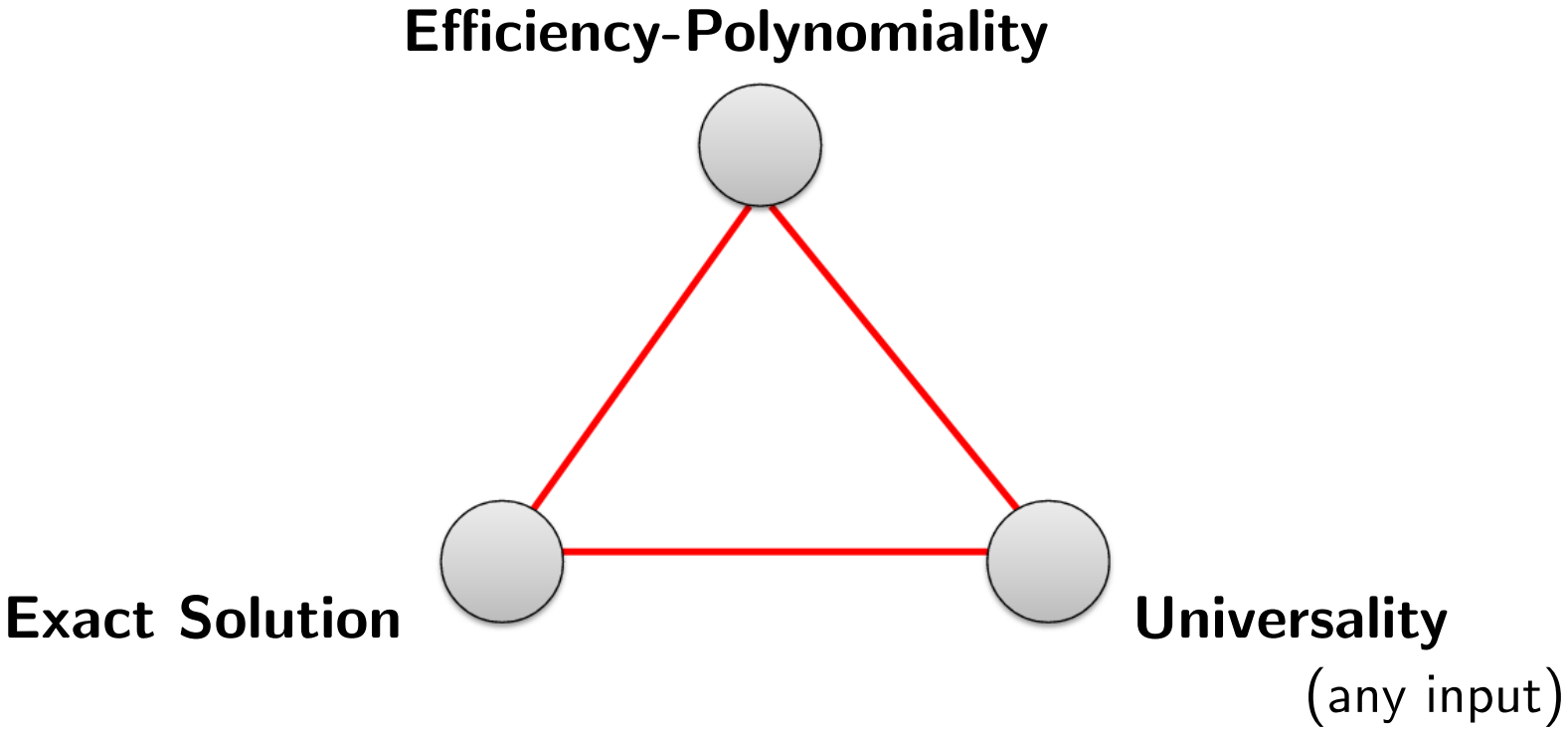
New Proof by Dinur (STOC 2006)

Motwani, Szegedi

90's: PCP and Approximation

Century 21: Conquering NP-hard problems

Mission Impossible: We can't solve them (a) in **polynomial time** (b) **exactly** and (c) for **all instances**.



Century 21: Conquering NP-hard problems

- Giving up condition (a):
  - $1.003^n \leq 1.5^n \leq 2^n \leq 5^n \leq n! \leq n^n$.
  - $n^{\log \log n} \leq n^{\log n} \leq n^{\log^{13} n} \leq n^n$.

- $GI \in QuasiP = DTIME[2^{polylogn}]$ (Babai)

Christofidis, Arora, Tardos, Shmoys, Williamson
Century 21: Conquering NP-hard problems

(a) in **polynomial time** (b) **exactly** and (c) for **all instances**.

- Giving up condition (b): **Approximation** Algorithms.

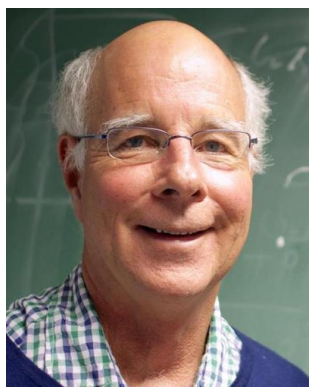




Johnson, Downey, Fellows, Courcelle
Century 21: Conquering NP-hard problems

- Giving up condition (c): Find large **subclasses** of the class of all instances for which the problem is solvable in **polynomial** time: e.g. HORNSAT
  - **Pseudo-Polynomial**, **Strongly Polynomial**.
  - **Parameterization**, *e.g.* VertexCover in $O(1.2738^k + kn)$ Parameterized Complexity ($2^k n^c$, n^k,...).

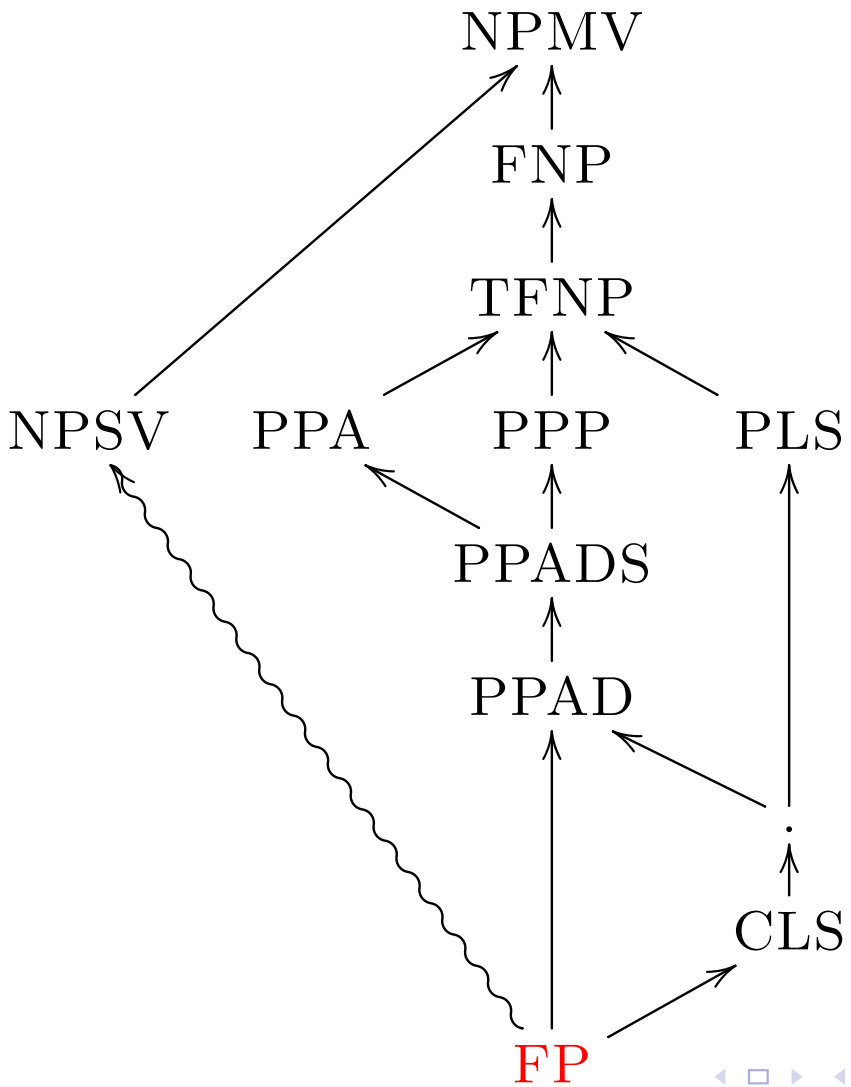





Courcelle's theorem: every graph property definable in the monadic second-order logic of graphs can be decided in linear time on graphs of bounded treewidth.

Papadimitriou, Yannakakis, Daskalakis

Century 21: Search Hierarchy



Papadimitriou, Yannakakis, Daskalakis

Century 21: Search Hierarchy

FP: e.g. find a perfect matching (*any*)

FNP: the class of partial multi-valued functions computed by an NPTM. The computation tree for input x has leaves which answer either **?** or the signature of the path y satisfying $R(x, y)$. e.g. find a clique of size $n/4$

NPMV: the class of partial multi-valued functions computed by an NPTM. The computation tree for input x has leaves which answer either **?** or one of the possible output strings

NPSV: single-valued NPMV functions, e.g., factoring

TFNP: FNP functions for which: $\forall x \exists y R(x, y)$. e.g. find a clique of size $n/4$, but you know there exists one, e.g., factoring

TFNP subclasses based on inefficient proof of existence

PLS: Polynomial Local Search, based on: *every finite directed acyclic graph has a sink*. e.g. find local optimum (e.g. POSNAE3FLIP, Pure Nash Equilibrium in general congestion games)

CLS: Continuous Local Search, **PLS** analogue for continuous spaces and functions. CLS contains search problems of local optimum approximation of a continuous function, using an oracle for a continuous function f.

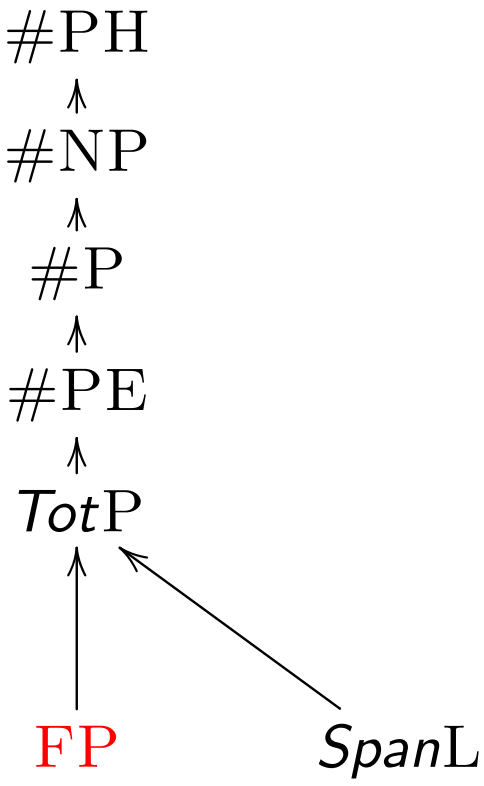
PPP:Polynomial Pigeonhole Principle, based on: *pigeonhole principle*

PPA: Polynomial Parity Argument, based on: *all graphs of max degree 2 have an even number of leaves* (e.g. given a Hamilton-cycle in an odd-degree graph find another one)

PPAD: Polynomial Parity Argument Directed. Like PPA, but graph is directed: find a sink or a source (e.g. Nash equilibrium, fixpoint theorems, Sperner's Lemma)

PPADS: Polynomial Parity Argument Directed Sink. Like PPAD, but find a sink

Hemaspaandra, Kolaitis, Pagourtzis, Z., Jerrum, Sinclair Goldberg Century 21: Counting



#QBF$_i$

#HAMILTON SUBGRAPHS

#SAT, #HAMILTONCYCLES

#SAT$_{+1}$

#PM, #DNF-SAT

#NONCLIQUES, #INDSETSALL

#RANKING

Hemaspaandra, Kolaitis, Pagourtzis, Z., Jerrum, Sinclair Goldberg Century 21: Counting

$\#\mathrm{PH} := (\#\mathrm{P})^{\mathrm{PH}}$

$\#\mathrm{NP} := (\#\mathrm{P})^{\mathrm{NP}}$

$\#\mathrm{PE} :=$ the subclass of $\#\mathrm{P}$ that contains functions with easy decision version

$\mathit{Tot}\mathrm{P} :=$ consists of all functions for which there exists a polynomial-time nondeterministic Turing machine (NPTM) such that the function value on x is equal to the total number of computation paths of M on input x

$\mathit{Span}\mathrm{L} :=$ # distinct outputs of a logspace NTM transducer

- $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{TotP}[1]}$

Century 21: Parameterized Complexity

Fixed Parameter Tractability (FPT): Solvability of problems in $O(f(k(x)) \cdot p(|x|))$ time, for some computable function f, and a parameter k.

The advantage of this approach is that we can concentrate the hardness of a problem to a certain parameter.

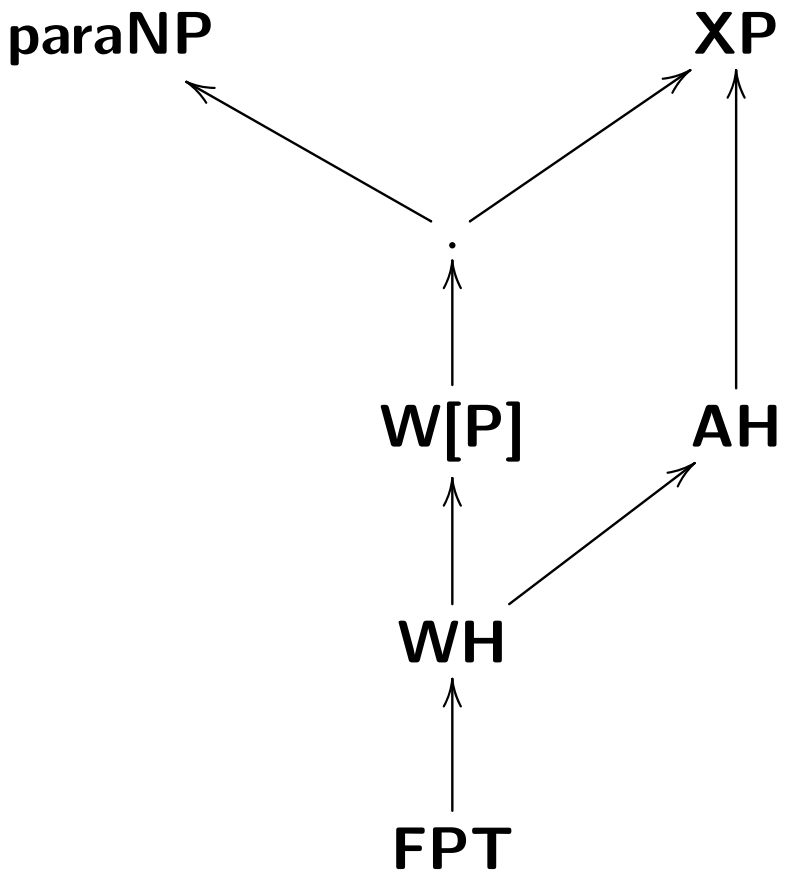
As in the theory of NP-completeness, there is the class W[P] of parameterized intractable problems.

ParaNP is the non-deterministic analogue of FPT.

XP is the parameterized analogue of EXP.

The question FPT vs W[P] is the parameterized analogue of P vs NP.

Century 21: Parameterized Complexity



Century 21: Non-Uniform Circuit Complexity

P/poly is the class of languages decided by a circuit family, such that each circuit has polynomial size, it properly contains P and BPP, but also undecidable problems.

Theorem (Karp-Lipton (1982))

If NP $\subseteq$ P/poly, *then* PH $= \Sigma_2^p$.

Theorem (Razborov-Andreev-Alon-Boppana (circa 1990))

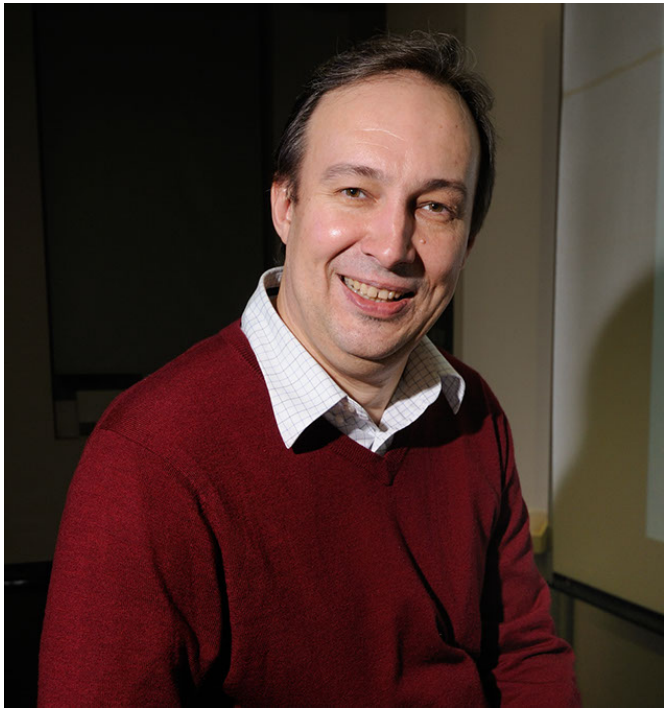
There exists an $\varepsilon > 0$, *s.t.* $\forall k \leq n^{1/4}$, *the k-clique problem cannot be decided by monotone circuits of size less than* $2^{\varepsilon\sqrt{k}}$.

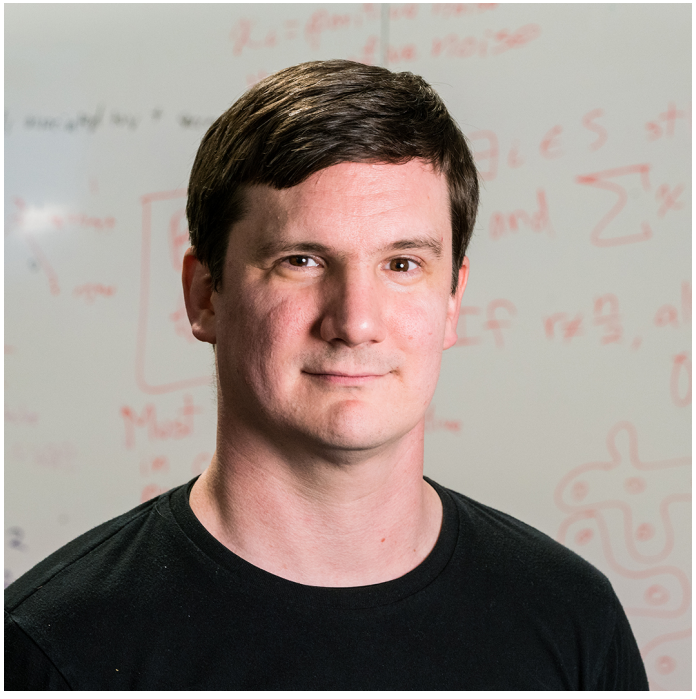
ACC^0 is the non-uniform analogue of AC^0, and we use also generalized parity (modulo) gates.

Theorem (Williams (2010))

$$\mathbf{NEXP} \not\subseteq \mathbf{ACC}^0$$

Non-Uniform Circuit Complexity (Razborov, Williams)





Century 21: Quantum Complexity

Quantum Complexity Theory adopts the quantum models of computation, such as Quantum TMs and Quantum Circuits.

The gates of such circuits are unitary transformations of their input.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i \cdot \frac{\pi}{4}} \end{pmatrix}$$

Quantum Complexity Classes like BQP (the quantum analogue of BPP), QMA (corresponding to MA), PQP (corresponding to PP) and QIP (corresponding to IP) appeared, and related with the classical model.

Century 21: Quantum Complexity

Theorem (Grover)

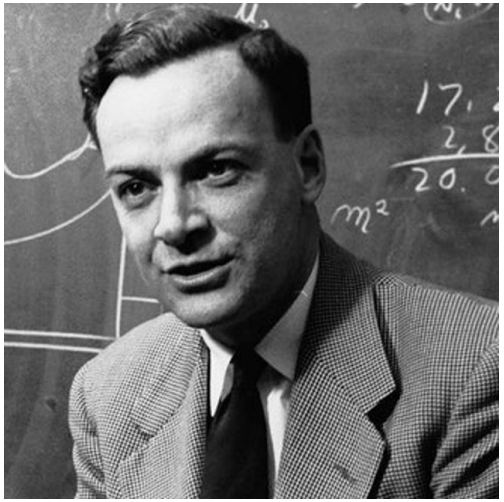
There is a quantum algorithm computing the position of an object s in a list of size N in $O(\sqrt{N})$ steps.

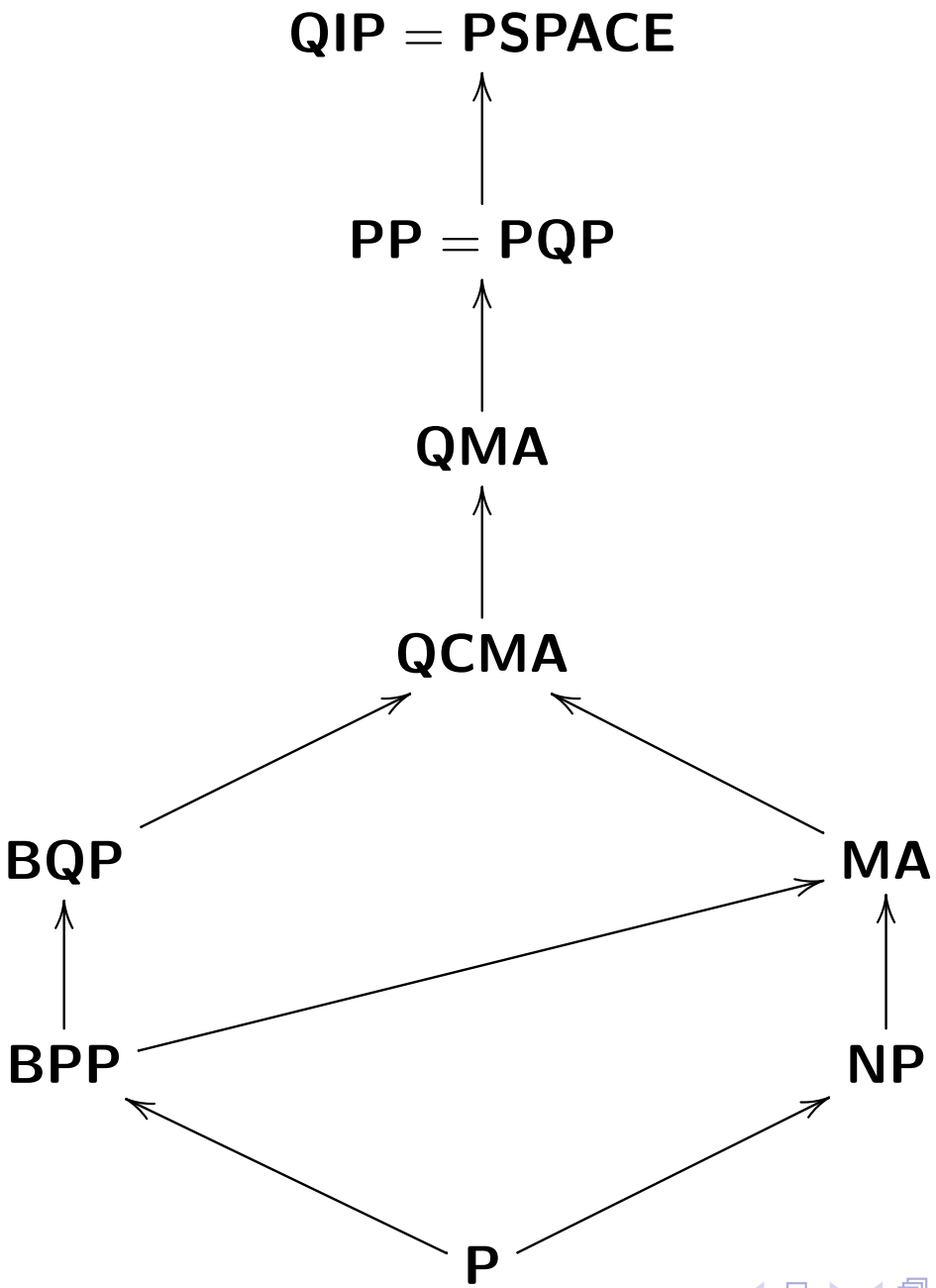
The well-known FACTORING problem is in **BQP** (Shor).

DISCRETE LOGARITHM is in **BQP**.

SUBGROUP NON-MEMBERSHIP (Given a subgroup $(H, \cdot)$ of a group $(G, \cdot)$, is a given $g \in G$ not in H?) is in **QMA**.

Quantum Complexity (Feynman, Shor, Grover, U. Vazirani)




QIP = PSPACE
PP = PQP
QMA
QCMA
BQP
MA
BPP
NP
P

Different ways of analyzing complexity

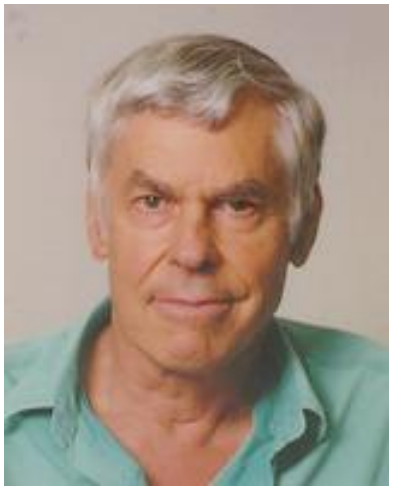
- **Worst-case Analysis**: Usually
- **Average-case Analysis**: Based on distribution of input instances
- **Best-case Analysis**: For cryptography, in order to avoid any attack.
- **Amortized Analysis**: Better performance for repeated actions by rearranging data.
- **Smooth Analysis**

Smooth Analysis

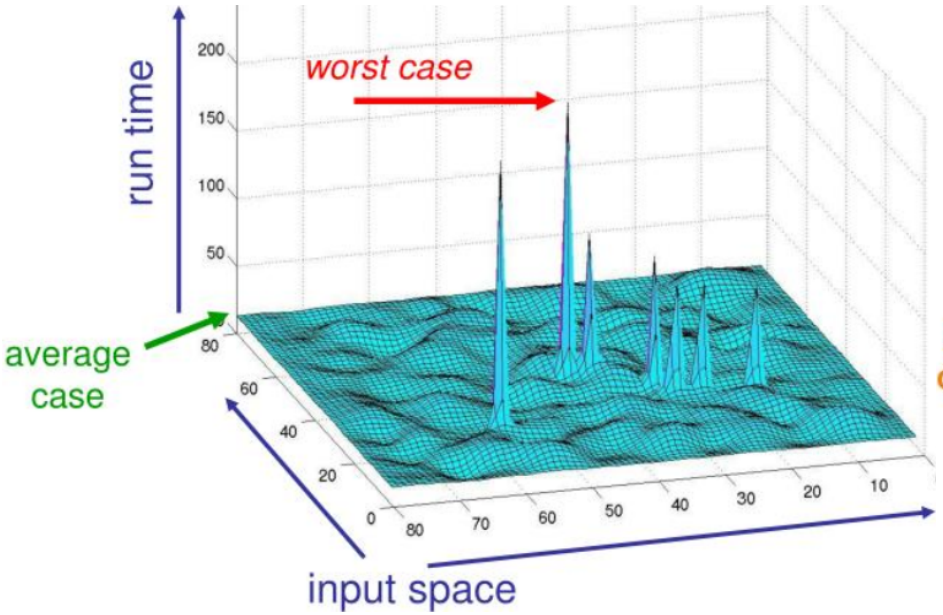
Shanghua Teng

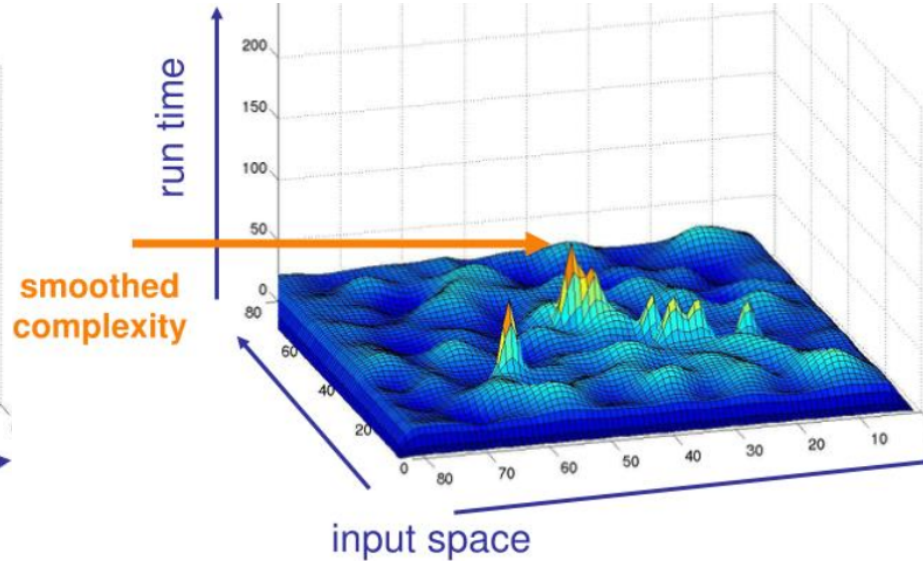
Daniel Spielman

Stephen Smale, Michael Shub, René Beier, Berthold Vöcking



Smooth Analysis





Illustrative example: Simplex (Dantzig)

Smooth analysis suggests that for problems where we have bad worst instances it is worthy to perturb first.

Thank You!