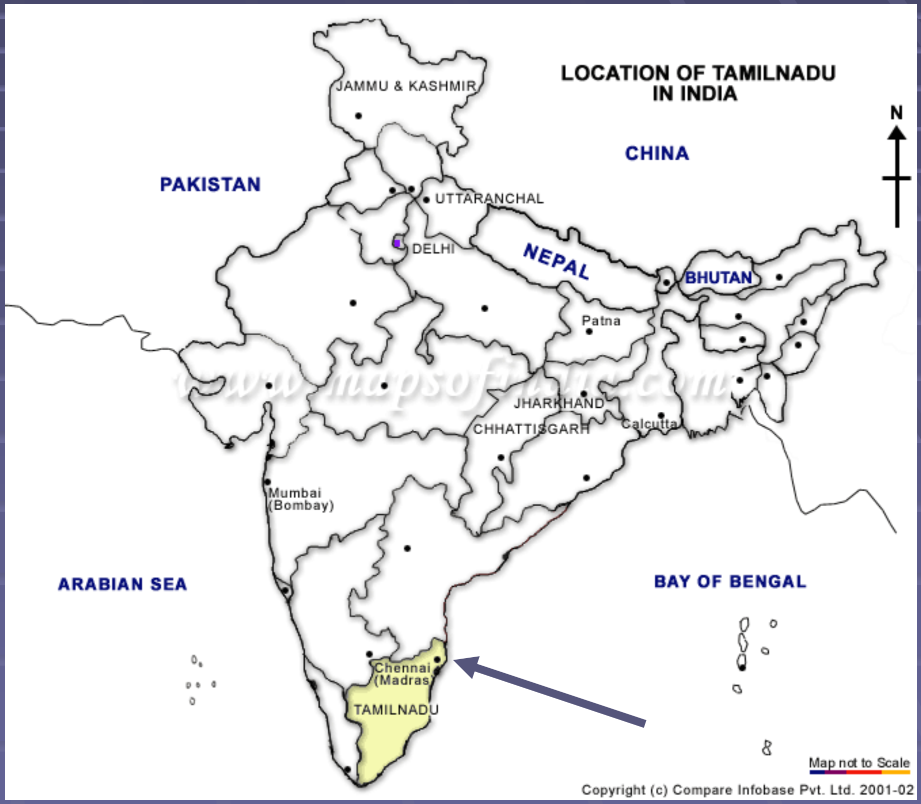
My Favorite Ten Complexity Theorems of the Past Decade II

Lance Fortnow
University of Chicago

Madras, December 1994

- Invited Talk at FST&TCS '04
- My Favorite Ten Complexity Theorems of the Past Decade



Why?

- Ten years as a complexity theorist.
- Looking back at the best theorems during that time.
- Computational complexity theory continually produces great work.
- Use as springboard to talk about research areas in complexity theory.
- Let’s recap the favorite theorems from 1985-1994.

Favorite Theorems 1985-94
Favorite Theorem 1

- Bounded-width Branching Programs Equivalent to Boolean Formula
  - Barrington 1989

Favorite Theorems 1985-94
Favorite Theorem 2

- Parity requires $2^{\Omega(n^{1/d})}$ gates for circuits of depth d.
  - Håstad 1989

Favorite Theorems 1985-94
Favorite Theorem 3

- Clique requires exponentially large monotone circuits.
  - Razborov 1985

Favorite Theorems 1985-94
Favorite Theorem 4

- Nondeterministic Space is Closed Under Complement
  - Immerman 1988 and Szelepcsényi 1988

Favorite Theorems 1985-94
Favorite Theorem 5

- Pseudorandom Functions can be constructed from any one-way function.
  - Impagliazzo-Levin-Luby 1989
  - Håstad-Impagliazzo-Levin-Luby 1999

Favorite Theorems 1985-94
Favorite Theorem 6

- There are no sparse sets hard for NP via bounded truth-table reductions unless P = NP
  - Ogihara-Watanabe 1991

Favorite Theorems 1985-94
Favorite Theorem 7

- A pseudorandom generator with seed of length $O(s^2(n))$ that looks random to any algorithm using $s(n)$ space.
  - Nisan 1992

Favorite Theorems 1985-94
Favorite Theorem 8

- Every language in the polynomial-time hierarchy is reducible to the permanent.
  - Toda 1991

Favorite Theorems 1985-94
Favorite Theorem 9

- PP is closed under intersection.
  - Beigel-Reingold-Spielman 1994

Favorite Theorems 1985-94
Favorite Theorem 10

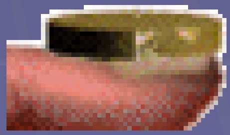
- Every language in NP has a probabilistically checkable proof that can be verified with O(log n) random bits and a constant number of queries.
  - Arora-Lund-Motwani-Sudan-Szegedy 1992

Kyoto, March 2005

- Invited Talk at NHC Conference.
- Twenty years in field.
- My Favorite Ten Complexity Theorems of the Past Decade II

Derandomization

- Many algorithms use randomness to help searching.
- Computers don’t have real coins to flip.
- Need strong pseudorandom generators to simulate randomness.

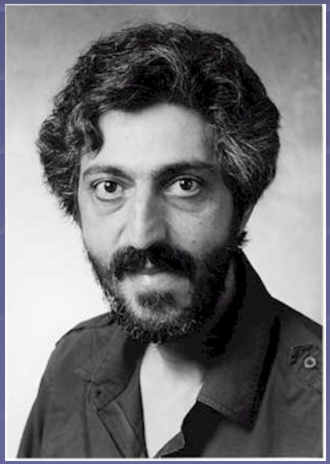


Hardness vs. Randomness

- BPP – Class of languages computable efficiently by probabilistic machines
- 1989 – Nisan and Wigderson
  - If exponential time does not have circuits that cannot solve EXP-hard languages on average then P = BPP.
- Many extensions leading to …

Favorite Theorem 1

- If there is a language computable in time $2^{O(n)}$ that does not have $2^{\varepsilon n}$-size circuits then P = BPP.
  - Impagliazzo-Wigderson '97



Primality

- How can we tell if a number is prime?

Favorite Theorem 2

- Primality is in P
  - Agrawal-Kayal-Saxena 2002

Complexity of Primality

- Primes in co-NP: Guess factors
- Pratt 1975: Primes in NP
- Solovay-Strassen 1977: Primes in co-RP
- Primality became the standard example of a probabilistic algorithms
- *Primality is a problem hanging over a cliff above P with its grip continuing to loosen every day.* – Hartmanis 1986

More Prime Complexity

- Goldwasser-Kilian 1986
- Adleman-Huang 1987
  - Primes in RP: Probabilistically generate primes with proofs of primality.
- Fellows-Kublitz 1992: Primes in UP
  - Unique witness to primality
- Agrawal-Kayal-Saxena – Primes in P

Division

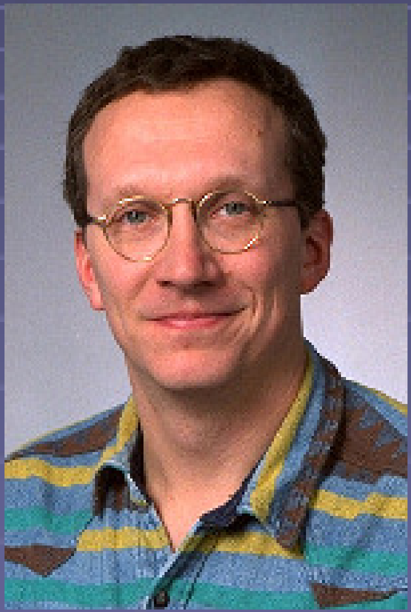
- Division in Non-uniform Logspace
  - Beame-Cook-Hoover 1986
- Division in Uniform Logspace
  - Chiu 1995
- Division in Uniform NC_1
  - Chiu-Davida-Litow 2001
- Division in Uniform TC_0
  - Hesse 2001

Probabilistically Checkable Proofs

- From 1994 list:
  - Every language in NP has probabilistically checkable proof (PCP) with O(log n) random bits and constant queries.
  - Arora-Lund-Motwani-Sudan-Szegedy
- Need to improve the constants to get stronger approximation bounds.

Favorite Theorem 3

- For any language L in NP there exists a PCP using O(log n) random coins and 3 queries such that
  - If x in L verifier will accept with prob ≥ 1-ε.
  - If x not in L verifier will accept with prob ≤ ½.
- Håstad 2001



Approximation Bounds

- Given a 3CNF formula we can find assignment that satisfies 7/8 of the clauses by choosing random assignment.
- By Håstad can’t do better unless P = NP.
- Uses tools of parallel repetition and list decodable codes that we will see later.

Connections

- Beauty in results that tie together two seemingly different areas of complexity.

Connections

- Beauty in results that tie together two seemingly different areas of complexity.
- Extractors – Information Theoretic

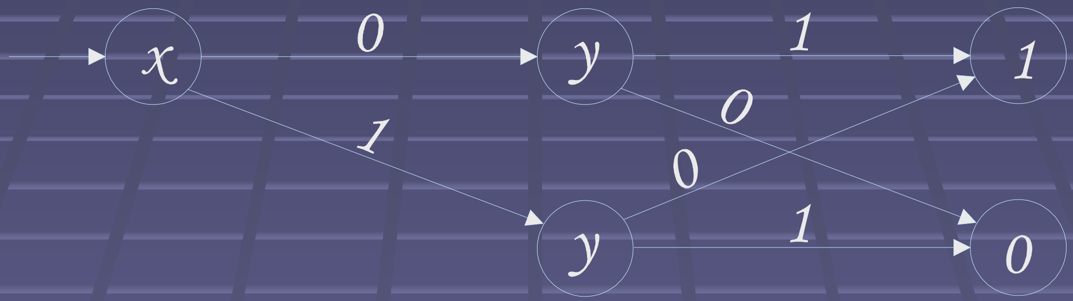
Connections

- Beauty in results that tie together two seemingly different areas of complexity.
- Extractors – Information Theoretic
- Pseudorandom Generators - Computational

Favorite Theorem 4

- Equivalence between PRGs and Extractors.
- Allows tools for one to create other, for example Impagliazzo-Wigderson to create extractors.
  - Trevisan 1999

Superlinear Bounds

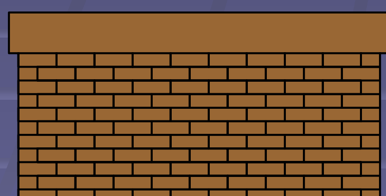


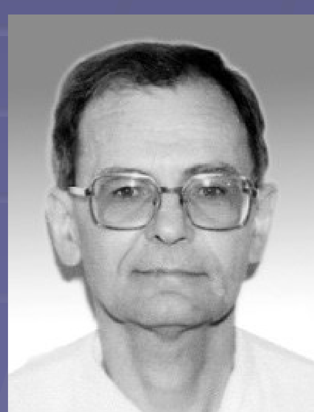
- Branching Programs
  - Size corresponds to space needed for computation.
  - Depth corresponds to time.
- We knew no non-trivial bounds for general branching programs.

Favorite Theorem 5

- Non-linear time lower bound for Boolean branching programs.
- Natural problem that any linear time algorithm uses nearly linear space.
- Ajtai 1999

Parallel Repetition





0110

1010

1100

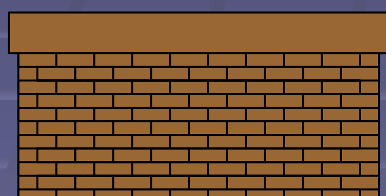
1001

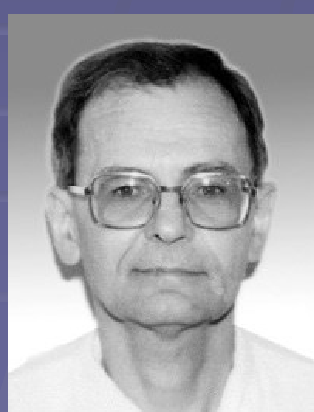
Accepts with prob ½

Parallel Repetition

0110 0010

1010 1011





1100 0100

1001 1011

Accepts with prob 1/4

Parallel Repetition

Favorite Theorem 6

- Parallel Repetition does reduce error exponentially in number of rounds.
- Useful in construction of optimal PCPs.
- Raz 1998

List Decoding

00101110

List Decoding

00101110

01000110010100111001010101010111001110111110001110

List Decoding

00101110

01000110010**100111**0010101010111001110111110001110

List Decoding

00101110

01000110010**100111**0010101010111001110111110001110

00101110

List Decoding

00101110

0100011**0010100111001010**1010111001110111110001110

List Decoding

00101110

0100011**0010100111001010**1010111001110111110001110

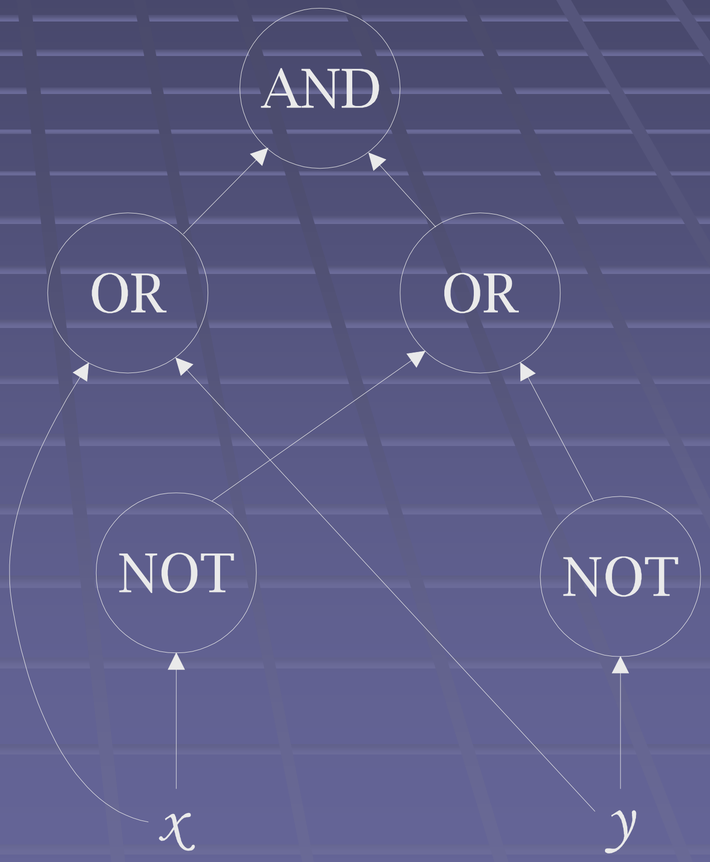
10010010
00101110
10111000
11101110

Favorite Theorem 7

- List Decoding of Reed-Solomon Codes Beyond Classical Error Bound
  - Sudan 1997
- Later Guruswami and Sudan gives algorithm to handle believed best possible amount of error.

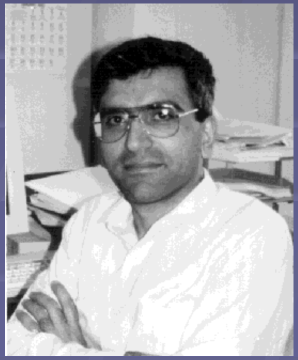
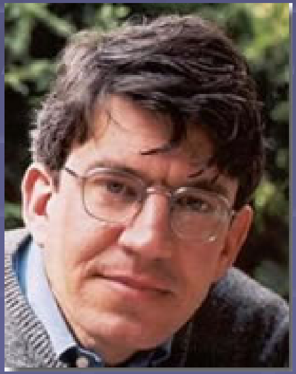
Learning Circuits

- Can we learn circuits by making equivalence queries, i.e., give test circuit and get out counterexample.
- No unless we can factor.



Favorite Theorem 8

- Can learn circuits with equivalence queries and ability to ask SAT questions.
  - Bshouty-Cleve-Gavaldà-Kannon-Tamon 1996


Corollaries

- If SAT has small circuits, we can learn circuit for SAT with SAT oracle.
- If SAT has small circuits then PH collapses to ZPP^{NP}.
  - Köbler-Watanabe

Quantum Lower Bounds

0100

10001

00010010

10001000

Quantum Lower Bounds

0100

10001

00010010

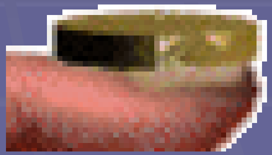
10001000

Favorite Theorem 9

- Razborov 2002
  - $N^{1/2}$ quantum bits required to compute set disjointness, i.e., whether the two strings have a one in the same position.
  - Matches upper bound by Buhrman, Cleve and Wigderson.

Derandomizing Space

- Given a randomized log n space algorithm can we simulate it in deterministic space?
- Simulate any randomized algorithm in $\log^2 n$ space.
  - Savitch 1969



Favorite Theorem 10

- Saks-Zhou 1999
  - Randomized log space can be simulated in deterministic space $\log^{3/2} n$.

Conclusions

- Complexity theory has had a great decade producing many ground-breaking results.
  - Every theorem builds on other work.
  - Wide variety of researchers from a cross section of countries.
- New techniques still needed to tackle the big separation questions.

The Next Decade

- Favorite Theorem 1
  - Undirected Graph Connectivity in Deterministic Logarithmic Space
  - Reingold 2005