15-251
Great Theoretical Ideas in Computer Science

Lecture 29:
Epilogue

December 8th, 2016

Goals of the course

- Learn about the foundational ideas and concepts in the theory of computation.

- Learn the mathematical constructs and techniques needed to understand and develop key computational concepts.

- Improve rigorous, logical, and abstract thinking skills.

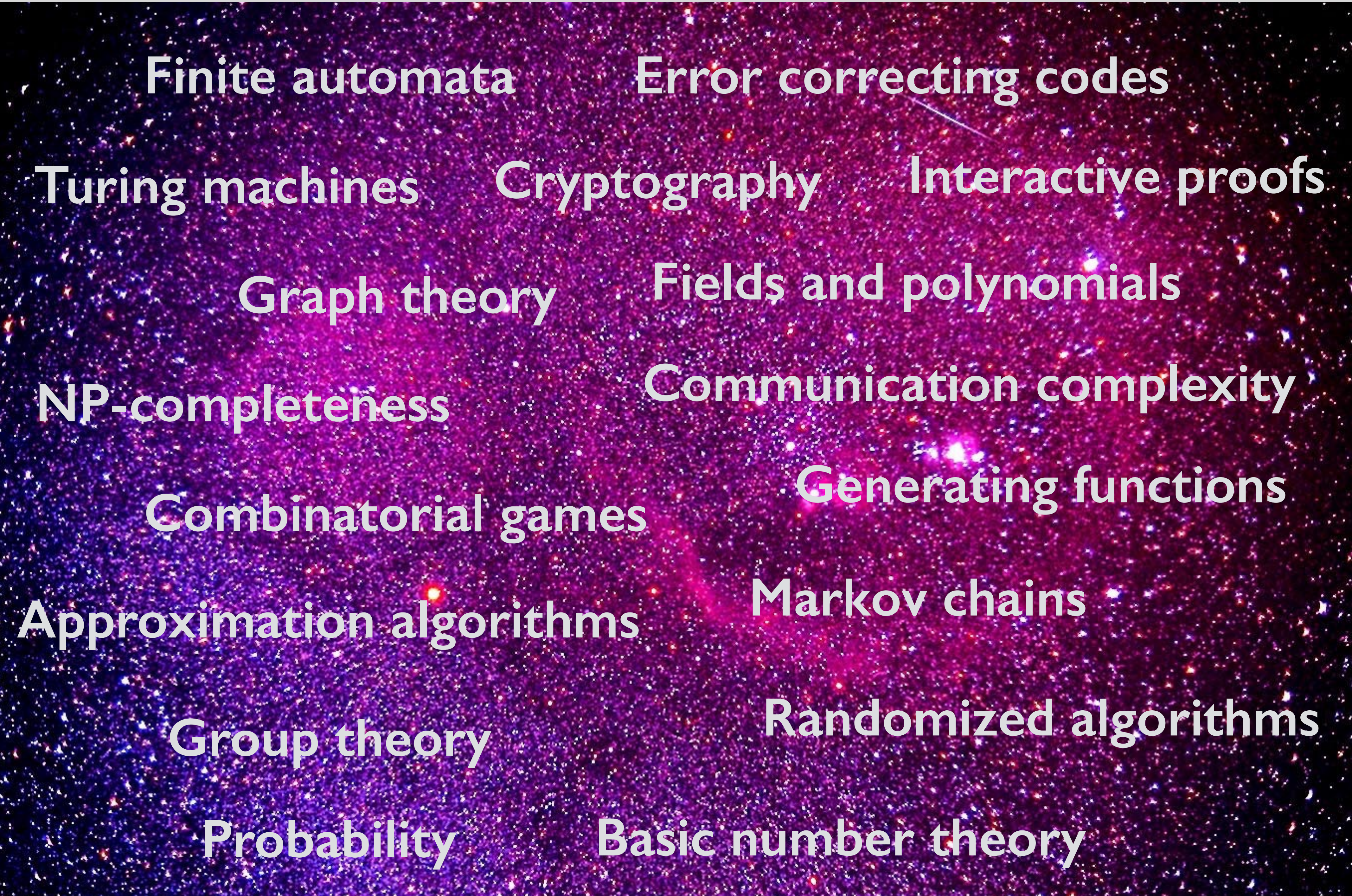
- Develop problem-solving skills.

- Refine proof-writing skills.

- Express complex ideas and arguments clearly, both in written and oral form.

- Cooperate with others in order to solve challenging and rigorous problems related to the study of computer science.

This is a “big picture” course



Topics we learned

- Formalization of computation (DFAs, TMs)
- Decidability/Undecidability
  (and relations to countability/uncountability)
- Computational complexity
  (and some interesting algorithms)
- **NP**-completeness and the **P** vs **NP** question
- Approximation algorithms
- Randomized algorithms

Topics we learned

- Gödel's incompleteness theorems
- Markov chains
- Cryptography
- Error-correcting codes
- Computer science perspective on proofs
- Communication complexity

Topics we learned

- Graph theory
- Probability theory
- Modular arithmetic
- Group theory
- Fields and polynomials
- Generating functions

Some big open questions

Relative power of resources

Resources: time, space, randomness, non-determinism.

Does non-determinism help
with respect to time efficient computation?

$$\mathrm{P} = \mathrm{NP}?$$

Relative power of resources

Resources: time, space, randomness, non-determinism.

Does non-determinism help
with respect to space efficient computation?

$$\mathsf{L} = \mathsf{NL}?$$

Relative power of resources

Resources: time, space, randomness, non-determinism.

Is time equivalent to space
with respect to efficient computation?

$$\text{P} = \text{PSPACE?}$$

Note:

$$\text{P} \subseteq \text{NP} \subseteq \text{PSPACE}$$

Relative power of resources

Resources: time, space, randomness, non-determinism.

Does randomness give us more power
with respect to time efficient computation?

$$\mathsf{P} = \mathsf{BPP}?$$

Interesting connection to circuit complexity:

certain circuit complexity lower bounds $\implies \mathsf{P} = \mathsf{BPP}$

$\mathsf{P} = \mathsf{BPP} \implies$ certain circuit complexity lower bounds

Relative power of resources

Resources: time, space, randomness, non-determinism.

Does randomness give us more power
with respect to time efficient computation?

$$\mathsf{P} = \mathsf{BPP}?$$

A major related result:

$$\text{PRIMES} \in \mathsf{P}$$

Relative power of resources

Resources: time, space, randomness, non-determinism.

Does randomness give us more power
with respect to space efficient computation?

$$\mathsf{L} = \mathsf{BPL}?$$

A major related result:

$$\text{USTCONN} \in \mathsf{L}$$

Relative power of resources

Resources: time, space, randomness, non-determinism.

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP} \subseteq \mathrm{NEXP}$$

Circuit complexity

Circuit complexity

<u>Circuits</u>: a clean and simple definition of computation.

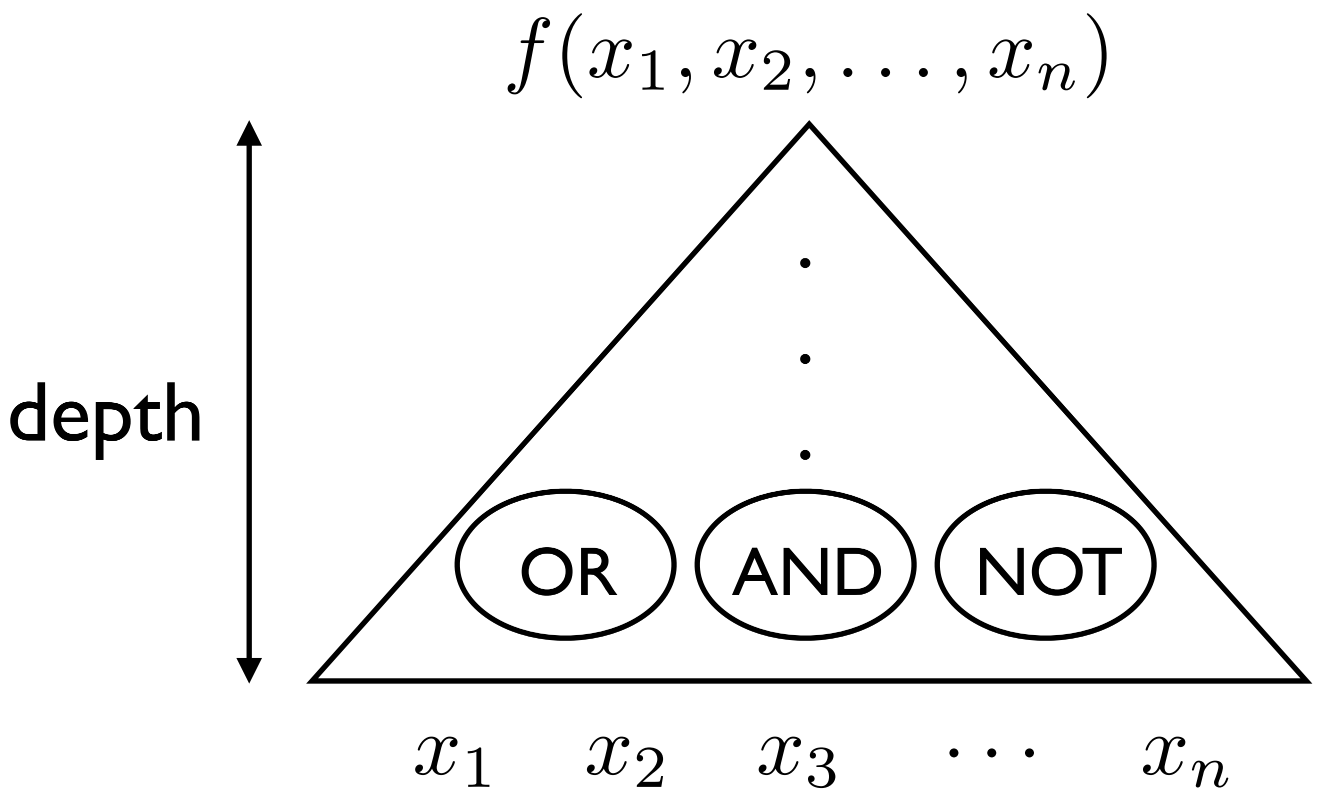
Just a composition of AND, OR, NOT gates.

poly-time TM $\implies$ poly-size circuits

no poly-size circuits $\implies$ no poly-time TM

So let's show SAT cannot be computed with poly-size circuits.

Circuit complexity

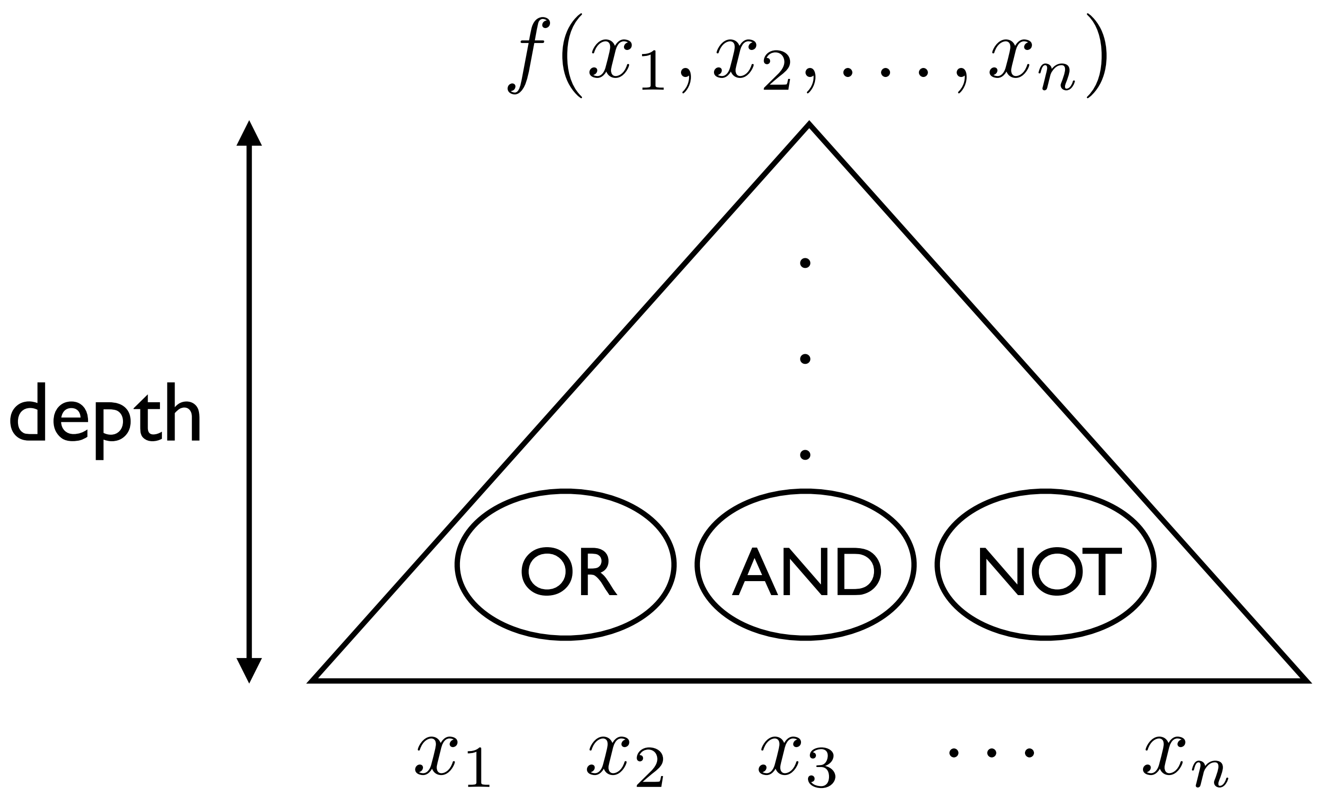


Let's restrict the circuit, make it less powerful.

What if we just allow constant depth?

Such circuits, in sub-exponential size, cannot compute parity function: $x_1 + x_2 + \cdots + x_n \pmod 2$

Circuit complexity

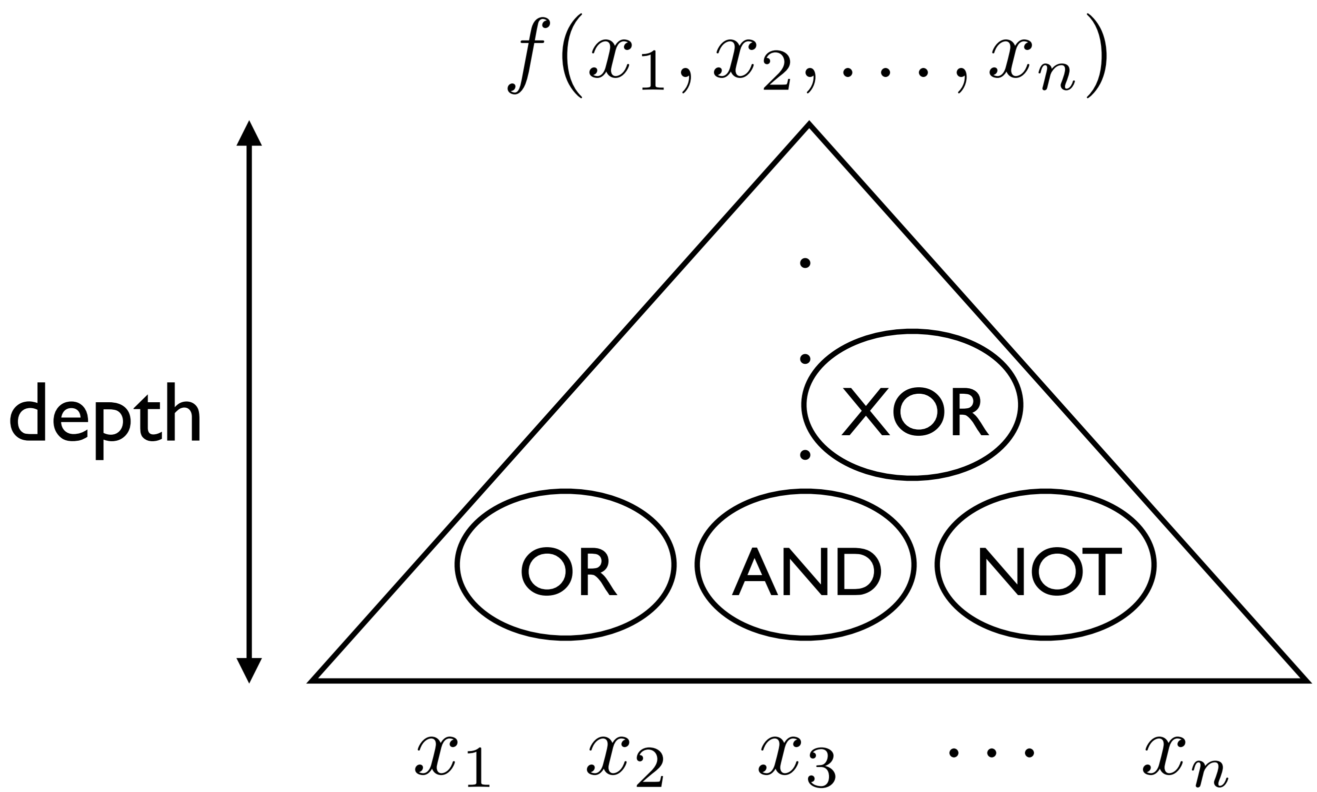


What if we just allow $O(\log n)$ depth?

parity **<u>can</u>** be computed in poly-size.

we can't prove lower bounds.

Circuit complexity



What if we just allow constant depth

but **add** parity gates to the circuit?

Circuit complexity

What if we just allow constant depth

but **add** parity gates to the circuit?

Such circuits, in polynomial size, cannot compute

$$\mathrm{mod}_3(x) = \begin{cases} 0 & \text{if } x_1 + x_2 + \cdots + x_n \equiv_3 0 \\ 1 & \text{otherwise} \end{cases}$$

Ok, let's add mod_3 gates to the circuit.

Or, instead of mod_2 and mod_3 gates,
just allow mod_6 gates.

Circuit complexity

Meanwhile…

Another restriction: remove NOT gates
(but no restriction on depth)

Alexander Razborov (1985):

Such poly-size circuits cannot compute CLIQUE.

We are so close to separating P and NP…

Circuit complexity

Alas…

Circuit complexity

Current frontier in circuit complexity:

Find a language in NP that cannot be computed by constant-depth, poly-size circuits with $\mathrm{and}, \mathrm{or}, \mathrm{not}, \mathrm{mod}_6$ gates.

In fact:

Find a language in NP that cannot be computed by **depth 3**, poly-size circuits with **just** mod_6 gates.

Circuit complexity

In fact:

Let's define a "generalized" mod6 gate.

For $A \subseteq \{0,1,2,3,4,5\}$

$$\text{mod}_6^A(x) = \begin{cases} 1 & \text{if } x_1 + x_2 + \cdots + x_n \pmod 6 \in A \\ 0 & \text{otherwise} \end{cases}$$

Find a language in NP that cannot be computed by **depth 2**, poly-size circuits with **just** "generalized" mod6 gates.

Please solve this problem!

Circuit complexity

Best known lower bound

For circuits with AND, OR, NOT gates:

Best known lower bound for an "explicit" function is

$$5n - \text{peanuts}$$

Circuit complexity

Another interesting type of circuit:

Circuits with threshold gates.

For $w_0, w_1, w_2, \ldots, w_n \in \mathbb{Z}$

$$\mathrm{thr}_w(x) = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 + \cdots + w_n x_n > w_0 \\ 0 & \text{otherwise} \end{cases}$$

Another major open problem:

Find a function that cannot be computed by poly-size, **dept-2** circuits composed of **only threshold gates**.

Circuit complexity

Why are circuit lower bounds so hard to prove?

1994

Steven Rudich
(CMU professor)

Alexander Razborov

Current techniques are unlikely to work!

"Natural Proofs barrier"

Algorithms

Algorithms

Matrix Multiplication

1978: $O(n^{2.796})$ by Pan

1979: $O(n^{2.78})$ by Bini, Capovani, Romani, Lotti

1981: $O(n^{2.522})$ by Schönhage

1981: $O(n^{2.517})$ by Romani

1981: $O(n^{2.496})$ by Coppersmith, Winograd

1986: $O(n^{2.479})$ by Strassen

1990: $O(n^{2.376})$ by Coppersmith, Winograd

2010: $O(n^{2.374})$ by Andrew Stothers (PhD thesis)

2011: $O(n^{2.373})$ by Virginia Vassilevska Williams

Algorithms

Matrix Multiplication

2014: $O(n^{2.372})$ by François Le Gall

2014: Ambainis, Filmus, Le Gall

These techniques are not going to let you go below $O(n^{2.3})$

Can we go down to $O(n^2)$?

Graph Isomorphism

Given two n-vertex graphs, are they isomorphic?

One of few problems not known to be in
P nor NP-complete.

Best known algorithm used to be: $2^{O(\sqrt{n \log n})}$

Now: $2^{O(\log^c n)}$

Algorithms

Factoring

Given a composite number, output a non-trivial factor.

One of few problems not known to be in
P nor NP-complete.

Best known algorithm: roughly $2^{O(n^{1/3})}$

There is a poly-time quantum algorithm.

Finding an n-bit prime

Given n, output a prime number with at least n digits.

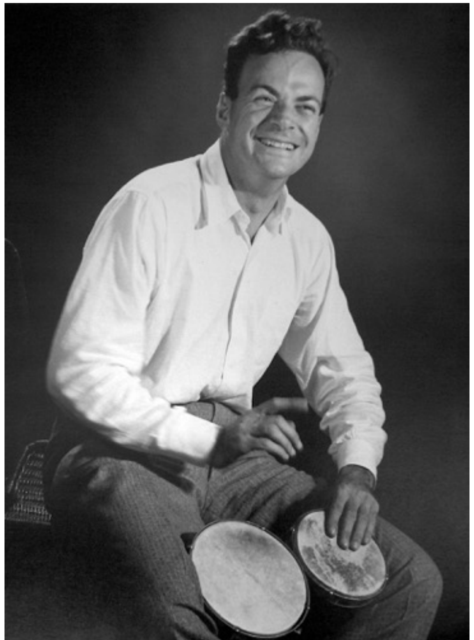
Find a poly(n) time deterministic algorithm.

poly(n) time randomized algorithm exists.

Quantum computation

Quantum computation

The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative.



-Richard Feynman

Quantum computation

BQP = quantum analog of BPP

$$\mathrm{BQP} = \mathrm{BPP}?$$

$$\mathrm{BQP} = \mathrm{NP}?$$

How are we going to tackle these tough questions?

Tackling math problems

(SOLO)

Andrew Wiles

Proved Fermat's Last Theorem 1995

(was open for 358 years)

Spent 7 years on it in secrecy.

Tackling math problems

(GROUP)

Paul Erdős

1913-1996

More than 500 collaborators

<u>Erdős number:</u>

degree of separation from Erdős

(he referred to children as "epsilons")

Tackling math problems

(OPEN)

Polymath projects:

Massively collaborative online mathematical projects

Gowers's Weblog

Mathematics related discussions

« A Tricki issue

Background to a Polymath project »

Is massively collaborative mathematics possible?

Of course, one might say, there are certain kinds of problems that lend themselves to huge collaborations. One has only to think of the proof of the classification of finite simple groups, or of a rather different kind of example such as a search for a new largest prime carried out during the downtime of thousands of PCs around the world. But my question is a different one. What about the solving of a problem that does not naturally split up into a vast number of subtasks? Are such problems best tackled by n people for some n that belongs to the set $\{1,2,3\}$? (Examples of famous papers with four authors do not count as an interesting answer to this question.)

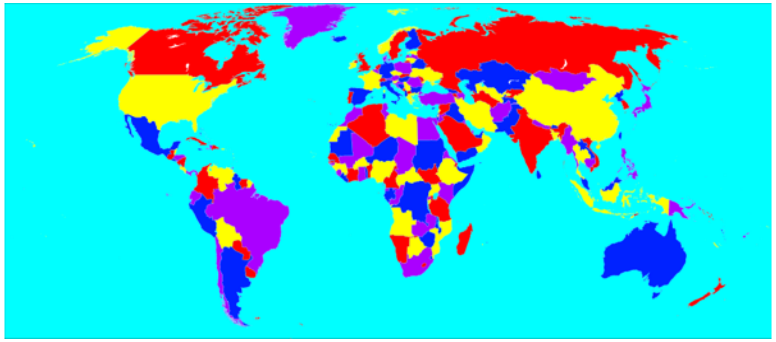
...

Timothy Gowers

Tackling math problems

(COMP)

4-Color Theorem



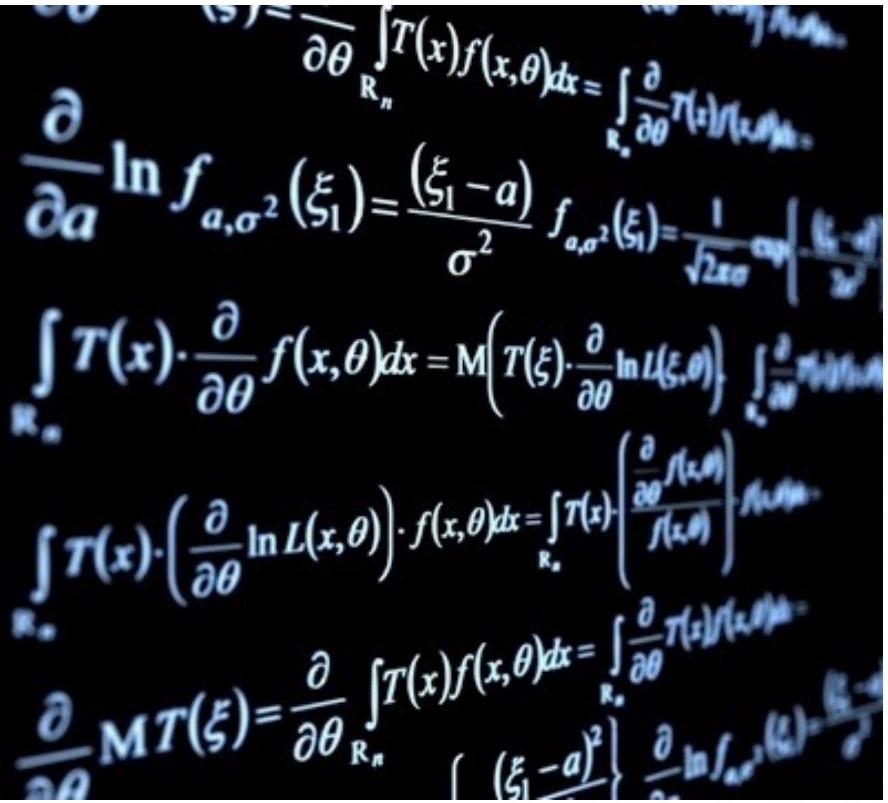
Reduce the problem to checking ~2000 cases.

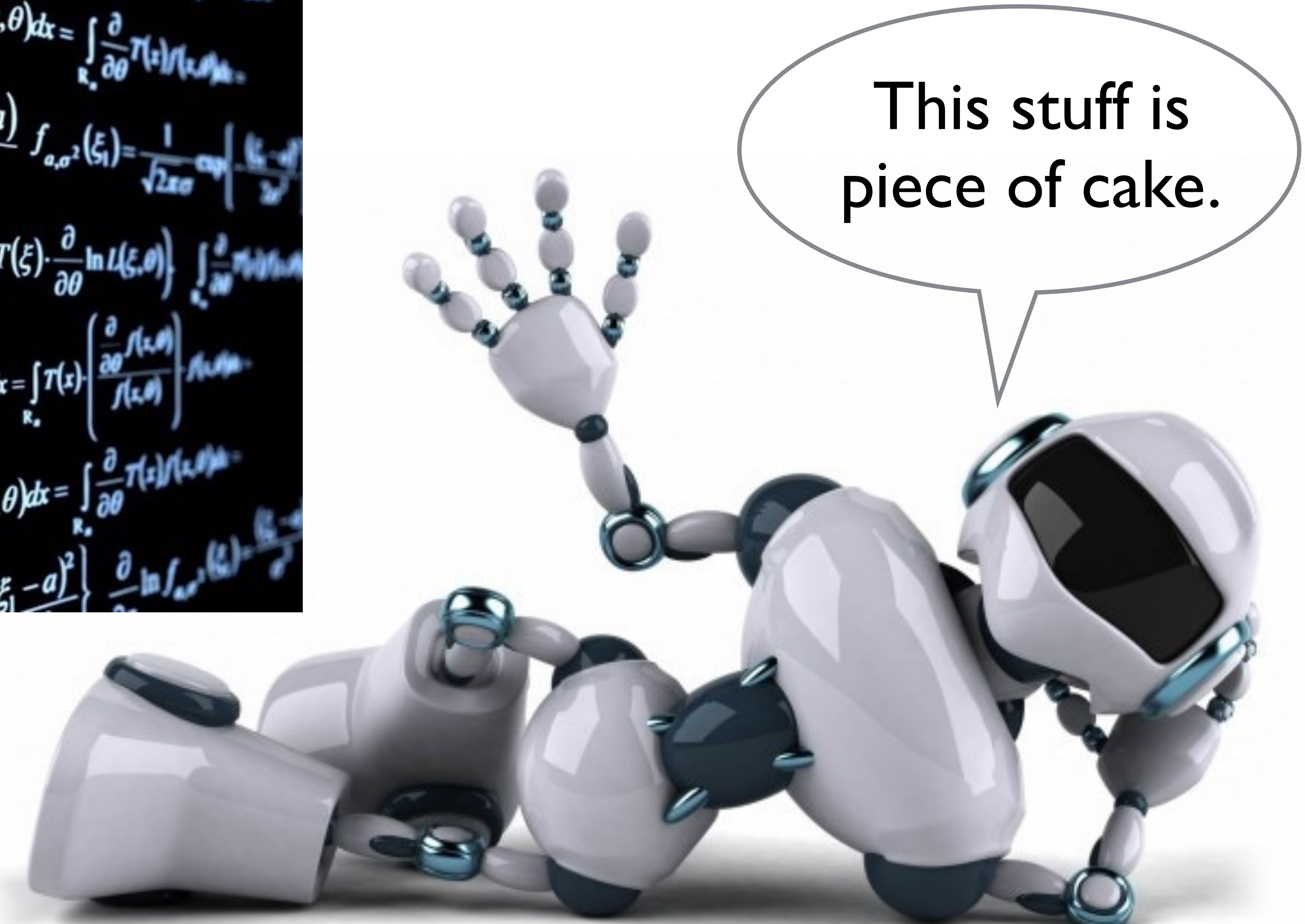
Let the machine check those cases.

Can expect more meaningful interactions between humans and computers in the future.

Tackling math problems

(SOLO FOR COMP)





Whatever the case may be, we need your help to make progress.

David Hilbert, 1900

The Problems of Mathematics

"Who among us would not be happy to lift the veil behind which is hidden the future; to gaze at the coming developments of our science and at the secrets of its development in the centuries to come? What will be the ends toward which the spirit of future generations of mathematicians will tend? What methods, what new facts will the new century reveal in the vast and rich field of mathematical thought?"


That's all Folks!