20 years of complexity

Highlights and emerging trends

20 years of GDR I(F)M

Sophie Laplante

cnrs | IRIF INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE | Université Paris Cité

What is complexity theory?

Algorithms

- Provide efficient algorithms, analyse their complexity

- Study algorithmic techniques and models

Complexity

- Study classes of problems with similar complexity and how they relate to one another (structural complexity) *e.g.* *P=NP?*

- Identify essential ressources and computational bottlenecks

What is complexity theory?

Complexity classes

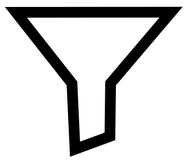
Computational problems

representative problems

Models

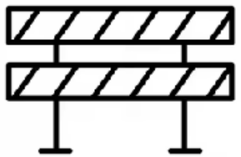
Resources

Tools



Bottlenecks

why some problems are harder than others



Barriers

obstacles to proving hardness

A brief history of complexity theory

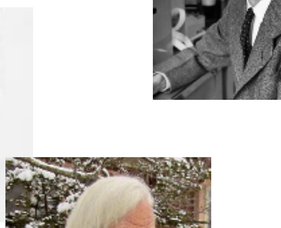
Early history

- *Entscheidungsproblem* (Hilbert & Ackermann 1928)

- **Computability** (Church & Turing 1936)
- Circuits and circuit size (Shannon 1949)

- *Gödel's letter to von Neumann (1956) asks about **efficiency** of first order logic proofs*
- "**Efficient computation**" (Jack Edmonds 1965)

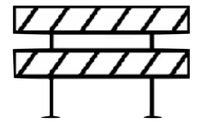
- **Space and time complexity**, diagonalization (Hartmanis & Stearns 1965)



The 70s : Key problems and models

- **P vs NP** (Cook & Levin 1971)
- Cook & Karp NP hard problems, SAT, **reductions** to compare problems
- Circuits, decision trees, algebraic models
- Logarithmic space (reachability, CVP)
- Randomized computation (1977)
- **The relativization barrier** (Baker Gill Solovay1975)

Trend : Techniques from **computability** *applied to efficient computation*



It is possible to prove P=NP and P ≠ NP **relative to an oracle**
Any proof of P vs NP cannot relativize

80s : Hardness and randomness

- Interactive proofs
- Complexity-based cryptography
- Hardness implies pseudorandomness
- Average-case derandomization
- Circuit lower bounds

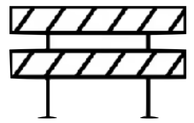
90s : Major advances, new barriers

Worst-case **derandomization**

- Error-correcting codes, randomness extractors,…

Interactive proofs, PCPs

- **Hardness of approximation**

Natural proofs (Razborov & Rudich) 

Quantum algorithms (Shor and Grover, 1994)

"constructive"
circuit lower bounds
imply breaking
cryptographical
assumptions

The past 20 years : methodology

2006-2026

- Breakthroughs
- Trends
- Conjectures
- Key problems
- Papers
- People

Hi Rahul,

I've been asked to give a talk on the highlights from the past 20 years in complexity theory. I'm asking around to see what other people think. Do you have a favorite result, or definition, or trend? Do you see an overarching theme emerging since 2005 or so? Pointers to some favorite papers that have inspired you perhaps?

All the best,

Sophie
--

✉ Major breakthroughs

Reingold's **Undirected connectivity** in log space

Ryan Williams **NEXP not in ACC0**

Ryan's recent result **DTIME(t(n)) in DSPACE(sqrt(t(n) log t(n)))**

Moser's proof of the **Lovasz local lemma**

✉️ Sources of inspiration

s-t conn is in logspace

Space-efficient **tree evaluation**

The **Unique Games Conjecture** and subexp time algorithm. Some of the most significant work in hardness of approximation in the last 20 years.

The **Strong Exponential Time Hypothesis** directly led to much of the early work on **fine-grained complexity**

Constant-depth **algebraic circuit lower bounds**

✉ Novel ideas

Meta-complexity and minimum circuit size problem

Circuit avoidance problem

Probabilistic Kolmogorov complexity

Algebraic circuits lower bounds

Progress on **derandomization of log-space**

Catalytic computation (my personal favourite) tree-evaluation, time t in space sqrt{t}, catalytic L=NL=BPL, algorithms for stconn.

✉ Recent trends

Meta-complexity. Algorithms from Natural Proofs, **characterization of one-way functions**,

Catalytic computation. Tree Evaluation and time T is in space close to sqrt{T}: a very surprising result to most complexity-theorists.

High-dimensional expanders, linear-rate **locally testable codes** with constant number of queries, **explicit lossless vertex expanders**

Korten "The Hardest **Explicit Construction**" played a role in the recent breakthrough **circuit lower bounds for exponential-time classes**.

Complexity-theoretic **derandomization** and the **Unique Games conjecture** (including the resolution of the 2-to-2 games conjecture).

MIP* = RE

Recurring themes

Circuit lower bounds

Derandomization

Hardness of approximation

Recent trends

- Meta-complexity
- Explicit constructions

Conspicuously missing

- Quantum complexity (except 1 mention of MIP*=RE)

Representative problems

Some complexity classes

L: O(log(n)) space on a Turing Machine (not including input string)

NL: nodeterministic log space

P: Polynomial time

NP: Nondeterministic poly time

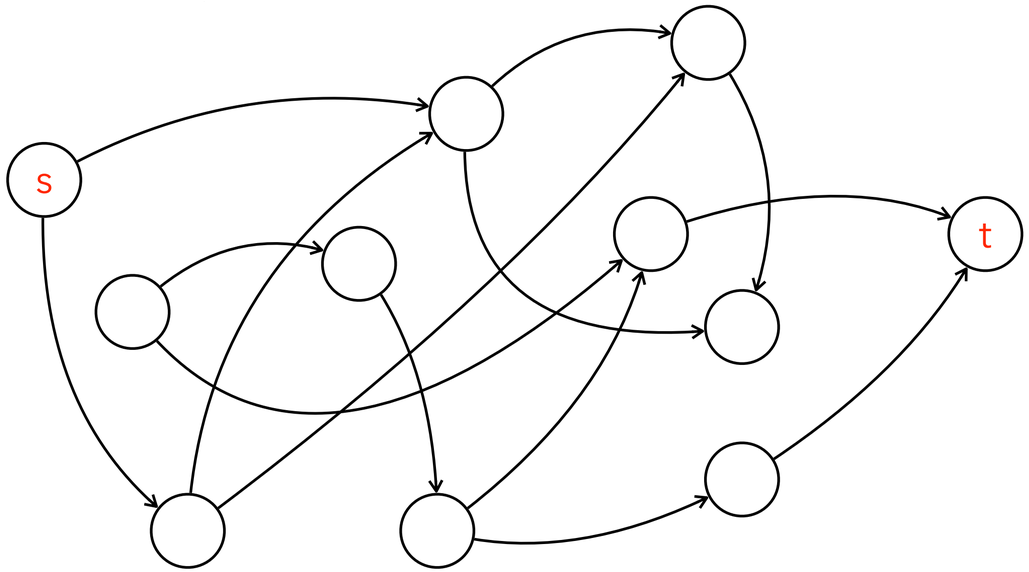
EXP: Exponential time

NEXP: Nondeterministic exponential time

DTIME(T(n)) : Deterministic time T(n)

Directed s-t connectivity

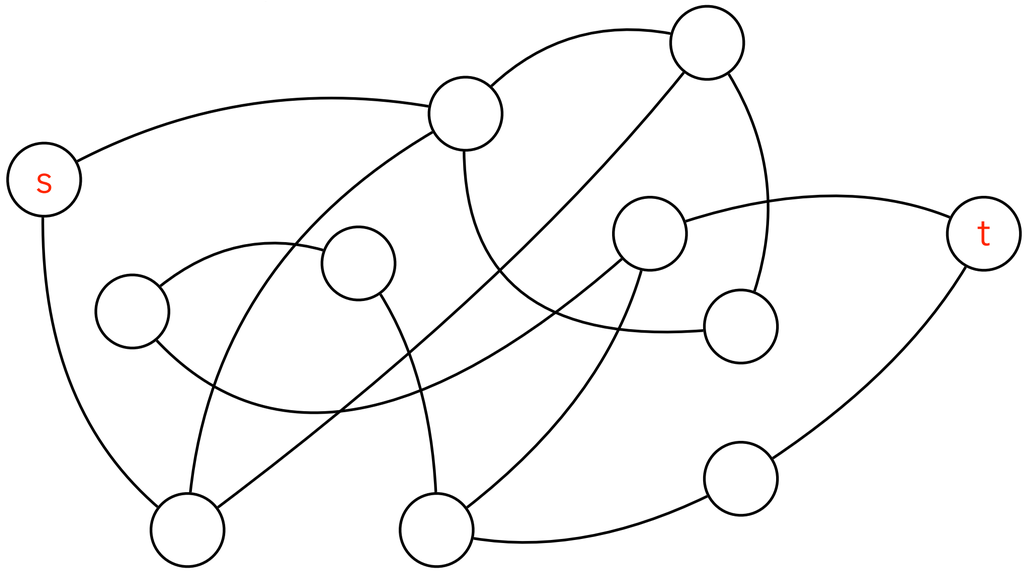
STCON : Given a **directed** graph G and two vertices s and t, decide if there is a path from s to t.



Complete for the class NL (**nondeterministic logspace**) – vertices are Turing Machine configurations, s = start config, t = accept config

Undirected s-t connectivity

USTCON : Given an **undirected** graph G and two vertices s and t, decide if there is a path from s to t.



Complete for the class **SL** (*symmetric* logspace: limited nondeterminism)

L ⊆ **SL**⊆ NL ⊆ DSPACE($\log(n)^2$) (Savitch 1970)

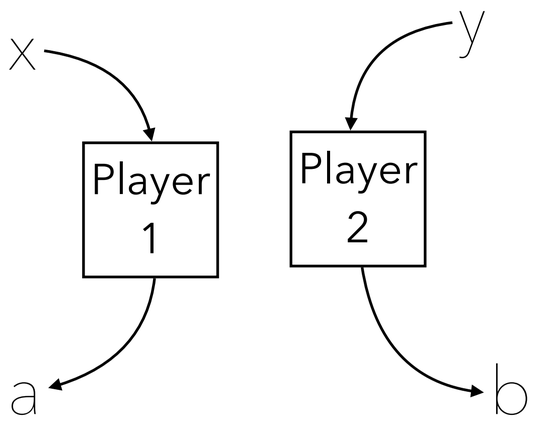
Other examples in this talk 

- Tree Evaluation Problem (P vs L?)
- Succinct3SAT (NEXP)
- MinCircuitSize (meta complexity)
- Range Avoidance Problem (derandomization, explicit constructions of combinatorial tools).

Conjectures & Hypotheses

Unique Games Conjecture

UG Conjecture [Khot 2002] : Approximating the optimal winning probability of a **unique game** is NP hard



Win if predicate **P**(a,b,x,y) is true

P is called **unique** if for any a, ∃! winning b

P≠NP + UGC implies **tight** **inapproximability** results for many problems (vertex cover, max cut, scheduling problems,…)

Exponential Time Hypothesis

ETH [Impagliazzo Paturi 1999]: 3SAT cannot be solved in **time $2^{o(n)}$**.

SETH : $\nexists$ constant **$c < 1$** s.t. $\forall$ $k \geq 3$ **kSAT** can be solved in **time $O(2^{cn})$**

- Parameter-preserving reductions between NP hard problems imply that assuming ETH, a plethora of **other problems also require 2^n time.**
- SETH implies **optimality of algorithms for graphs** of bounded treewidth [Lokshtanov, Marx, Saurabh 2010] and many others.
- Hugely influential e.g. in **fine-grained** complexity and **parameterized complexity**

Sensitivity conjecture

Definition [Cook Dwork 1982] For a boolean function **f**, an input **x** is **sensitive** at index **i** if $f(x) \neq f(x^{(i)})$ where $x^{(i)}$ is obtained from i by flipping the ith bit of x.

Example OR function, the input 00000 is sensitive on all 5 indices

Proposition if f has an input sensitive at **s** indices, then its **decision tree** complexity **D** verifies $s \leq D$

Conjecture [Nisan Szegedy 1994] $\exists$ c s.t. for every f $D \leq s^c$

Theorem [Huang 2019] $D \leq s^6$

Open $D \leq s^c$ for $c<6$. *

*Best known separation is a function for which $D \geq s^3$

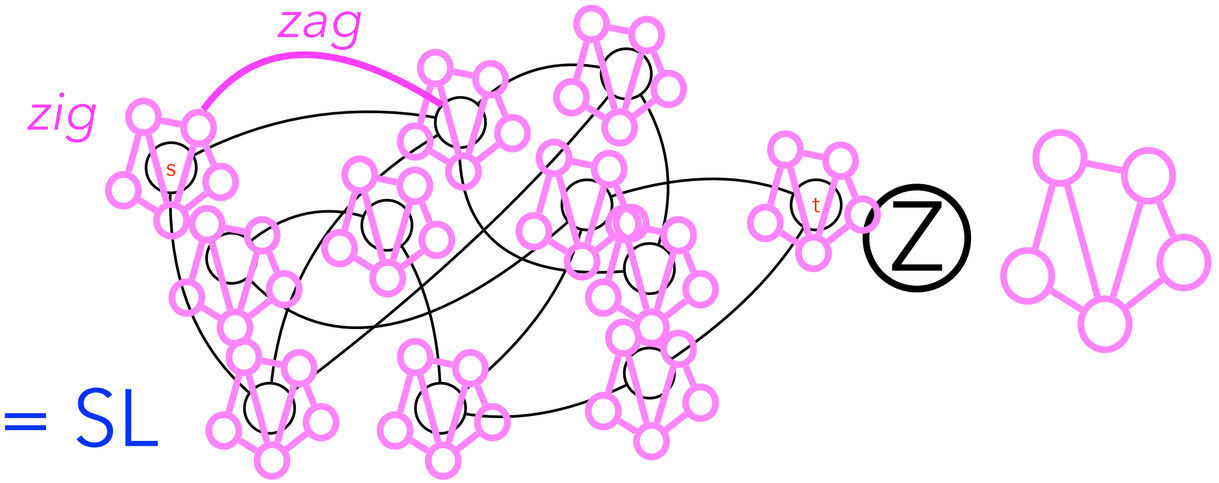
Breakthroughs

Undirected connectivity in log space

Random walk [Aleliunas, Karp, Lipton, Lovasz, Rackoff 1979]

- $L \subseteq SL \subseteq RL$

- $RL \subseteq L^{3/2}$ [Saks Zou 1999]



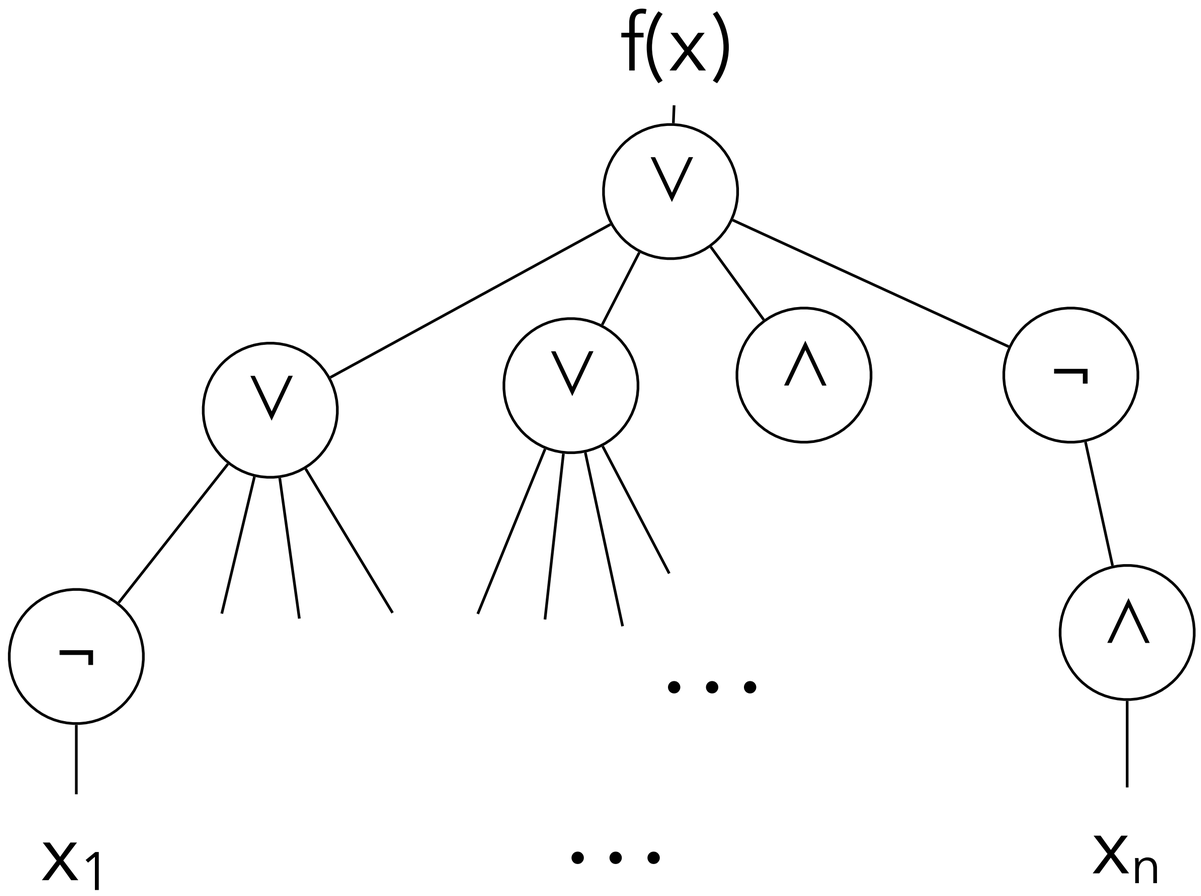
Theorem [Reingold 2004] $L = SL$

- Idea: USTCON on an expander graph is easy, so first improve expansion of G using zig zag product … then derandomize

Open : RL=L?

Circuit lower bounds

	fan-in/arity	*gates*	*depth*	*size*
NC^0	constant	$\vee \wedge \neg$	O(1)	poly
AC^0	unbounded	$\vee \wedge \neg$	O(1)	poly
ACC^0	unbounded	$\vee \wedge \neg$ **mod m**	O(1)	poly
TC^0	unbounded	$\vee \wedge \neg$ **MAJ**	O(1)	poly



$$NC^0 \subsetneq AC^0 \subsetneq ACC^0 \subseteq TC^0$$

(OR) (Parity) (Majority?)

Circuit lower bounds

constant depth

+ mod gates

+ MAJ gates

log(n) depth

log(n) space

$$NC^0 \subsetneq AC^0 \subsetneq \mathbf{ACC^0} \subseteq TC^0 \subseteq NC^1 \subseteq L \subseteq P \subseteq NP \subseteq EXP \subseteq \mathbf{NEXP}$$

OR

Parity

Majority

Matrix multiplication, rank,…

STCON

Theorem [Williams 2010] NEXP contains a problem not in (non-uniform) $\mathbf{ACC^0}$

Circuit lower bounds: useful problems


P

Circuit Value Problem (CVP)

Input: Circuit on n variables, input string x

Evaluate the circuit on x


NP

CIRCUIT-SAT

Input: Circuit on n variables

Decide if there exists an assignment to the variables that makes the circuit output TRUE


NEXP

SUCCINCT 3-SAT

Input: Circuit that **encodes a formula** on **2^n variables** and **2^n clauses**

Decide if the formula is satisfiable

Circuit lower bounds

Proposition $\exists\ L \in \mathsf{NTIME}[2^n] \setminus \mathsf{NTIME}[2^n/n^k]$

- by hierarchy theorem – diagonalization (1970's)

Thm [Williams 2010] : $\mathsf{NEXP} \subseteq \mathsf{ACC}^0$ implies $L \in \mathsf{NTIME}[2^n/n^k]$

- $L \leq$ succinct3SAT
- succinct3SAT $\leq \mathsf{ACC}^0$circuitSAT (using CVP $\in \mathsf{NEXP} \subseteq \mathsf{ACC}^0$)
- ACC^0-circuitSAT in $\mathsf{NTIME}[2^n/n^k]$ via dynamic programming

Circuit lower bounds

Proposition $\exists\, L \in \mathrm{NTIME}[2^n] \setminus \mathrm{NTIME}[2^n/n^k]$

- Thr [illegible] $2^n/n^k]$
- [illegible]
- [illegible] $C^0)$
- ACC^0-circuitSAT in $\mathrm{NTIME}[2^n/n^k]$ via dynamic programming

How does this work get around the barriers? There are several well-known barriers to proving lower bounds, and any proof claiming a new lower bound should try to explain a bit about how it manages to work around them. Briefly, the approach of this paper circumvents the natural proofs barrier because of its use of diagonalization: a proof by contradiction is given which appeals to a time hierarchy theorem, so constructivity is violated. (Furthermore, to our knowledge there is little evidence that ACC contains pseudorandom function generators anyway, so it's not clear that natural proofs should be considered a barrier for ACC.) Informally, the approach circumvents relativization and algebrization because it relies on an efficient ACC satisfiability algorithm, which uses non-relativizing properties of ACC circuits (reduction to a SYM^+ circuit). In general, the approach of using SAT algorithms to prove lower bounds appears fruitful for circumventing oracle-based barriers, because all known improved satisfiability algorithms break down when oracles (or algebraic extensions thereof) are added to the instance. That is, significant improvements over exhaustive search necessarily exploit structure in instances that black-box methods cannot see.

New and emerging trends

Metacomplexity

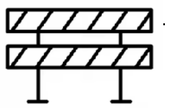
> “There’s this intuition that ‘oh, because P ≠ NP, it is difficult to prove that P ≠ NP,’” Williams said. “But in order to even make sense of this intuition, you’ve got to start thinking about the task of proving something like P ≠ NP as a computational problem.”

Quanta Magazine, Complexity Theory’s 50-Year Journey to the Limits of Knowledge

Metacomplexity

Computational complexity ***of*** complexity measures

Min Circuit Size Problem (**MCSP**)
Input: the **truth table** of a function
Compute size of **smallest circuit** for this function



If **MCSP** is NP-hard under deterministic "natural" reductions, then EXP does not contain P/poly [Kabanets Cai 2000]

(Partial-function) **MCSP*** is NP hard to approximate [Harihara 2022]
Proof uses a randomized reduction to avoid barrier

Metacomplexity

Computational complexity of complexity measures

Time bounded Kolmogorov complexity (K^t)
Input: a string x
Compute the length of a shortest program that prints x within time t(|x|)

Characteriziation of existence of one-way functions [Liu and Pass 2020]

The sleeper hit

Catalytic space

Reinventing space complexity [Buhrman Cleve Koucky Loff Speelman 2014]

Usual assumption: **memory is blank** at the start of the computation. It can be restored to its blank state at the end.

Catalytic-space computation:

- Small amount of blank space s(n)
- Additional $2^{s(n)}$ **"used" space** that should be restored at the end.

Catalytic space

Reversible computation means computation can be "undone" and auxiliary tape restored. *Transparent* computation is reversible when starting from arbitrary content of the tape

CL : catalytic logspace (s(n) = O(log(n))
CNL : catalytic nondeterministic logspace
CPrL : catalytic randomized logspace (correctness > 1/2)

[BCKLS 2014] $L \subseteq NL \subseteq TC^1 \subseteq CL \subseteq ZPP$
[Koucký, Mertz, Pyne, Sami 2025] CL =CNL=CPrL

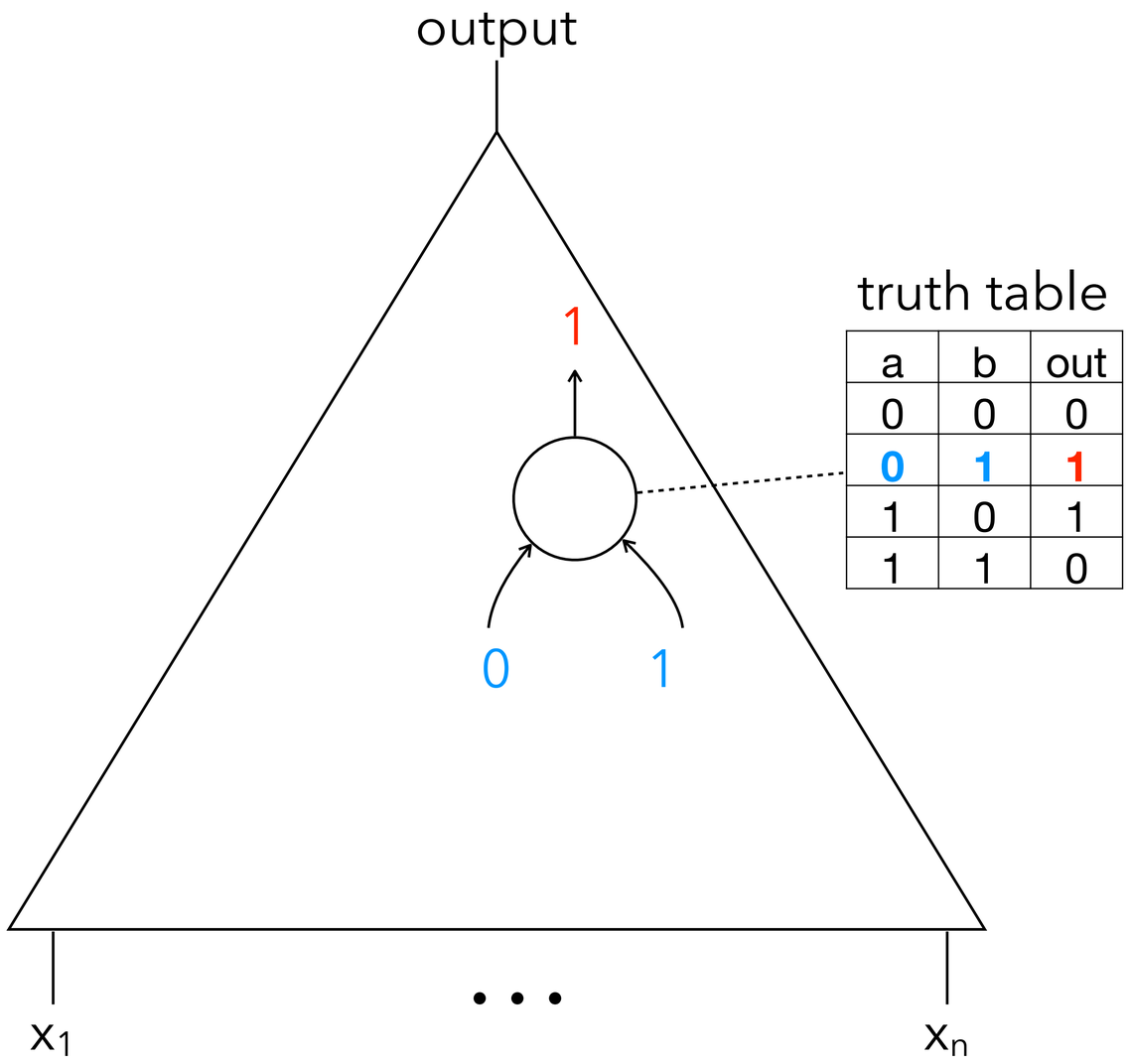
Catalytic space

Tree evaluation problem [S. Cook McKenzie Wehr Braverman Santhanam 2010]

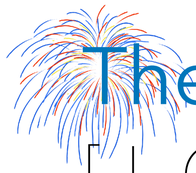
Input: tree with leaves labeled with variables and nodes with a truth table

Output: evaluate each node using truth table, and output the value at the root

Candidate to separate P from logspace using **composition**



Catalytic space



Theorem Tree Evaluation Problem is in space **log n log log n**
[J. Cook Mertz 2024]

Proof uses **catalytic space** techniques to save space.(and many other tricks and algebraic techniques)

Still open: TEP $\in$ **L**?

Catalytic space

Thm Tree Evaluation Problem is in space **log n log log n** [J. Cook Mertz 2024]

Thm [Williams 2025] Any computation using **time T** can be simulated in $\sqrt{Tlog(T)}$ space !

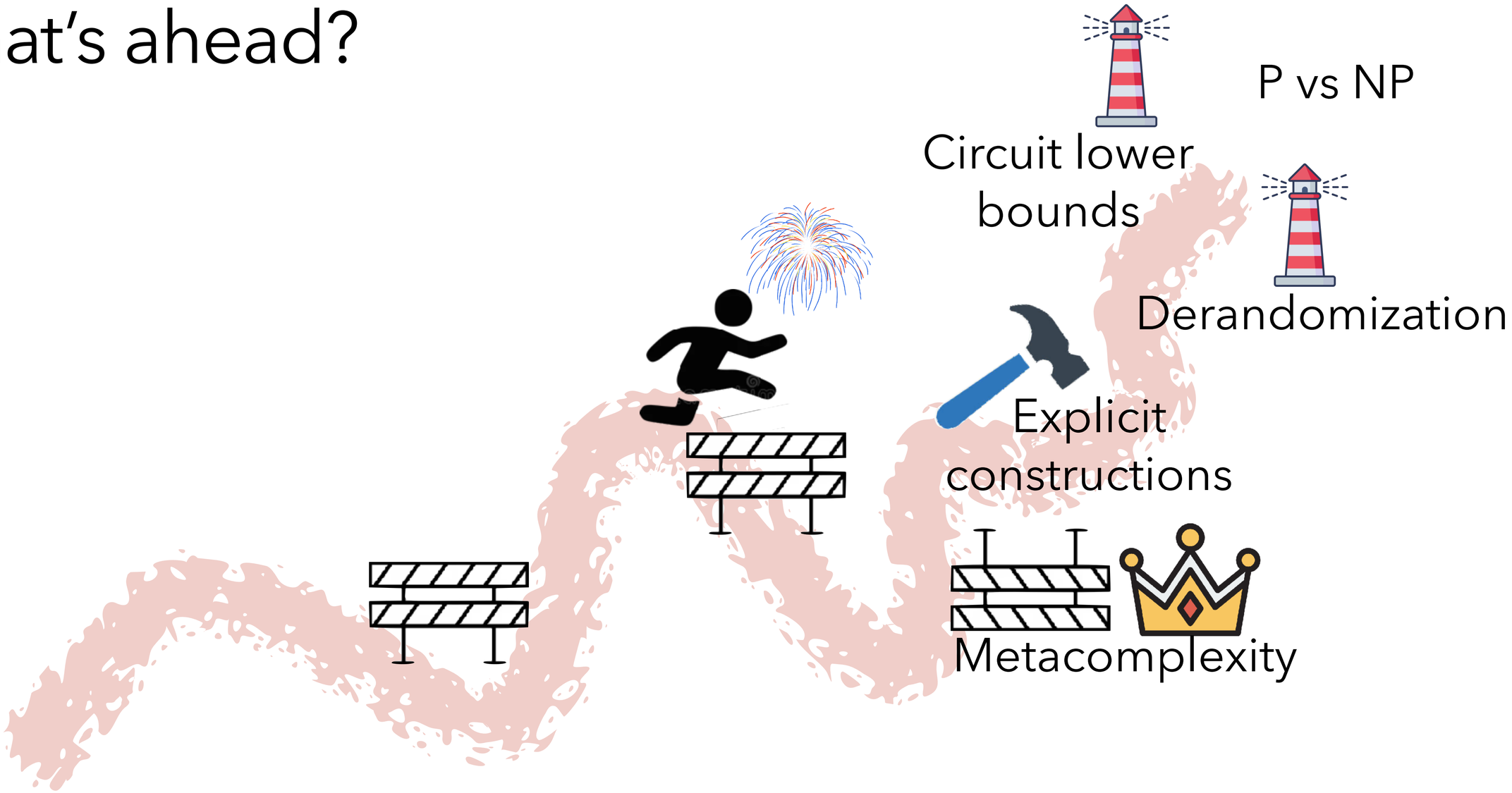
Previous bound was T/log(T) [Hopcroft, Paul, Valiant 1975]

Proof uses reduction(s) to Tree Evaluation Problem

There is more!

- Advances in **derandomization**
- **Algebraic circuit lower bounds** [Limaye Srinivasan Tavenas 2022]
- **Extended formulations** : why LPs cannot show P=NP [Fiorini Massar Pokutta Tiwari deWolf 2012]
- Adversary bound = **quantum query complexity** [Reichardt 2011]
- QIP=IP=PSPACE [Jain Ji Zhengfeng Upadhyay Watrous 2010]
- MIP*= RE [Ji, Natarajan, Vidick, Wright, Yuen 2020]
- ...

What's ahead?


P vs NP
Circuit lower bounds
Derandomization
Explicit constructions
Metacomplexity

Selected further reading

Quanta Magazine, Complexity Theory's 50-Year Journey to the Limits of Knowledge

Michal Koucký Bulletin of the EATCS Complexity Column
https://bulletin.eatcs.org

Lance Fortnow Favorite theorems, the complete list
https://blog.computationalcomplexity.org/2024/12/

Valentine Kabanets, Derandomization: A Brief Overview

Thanks for listening