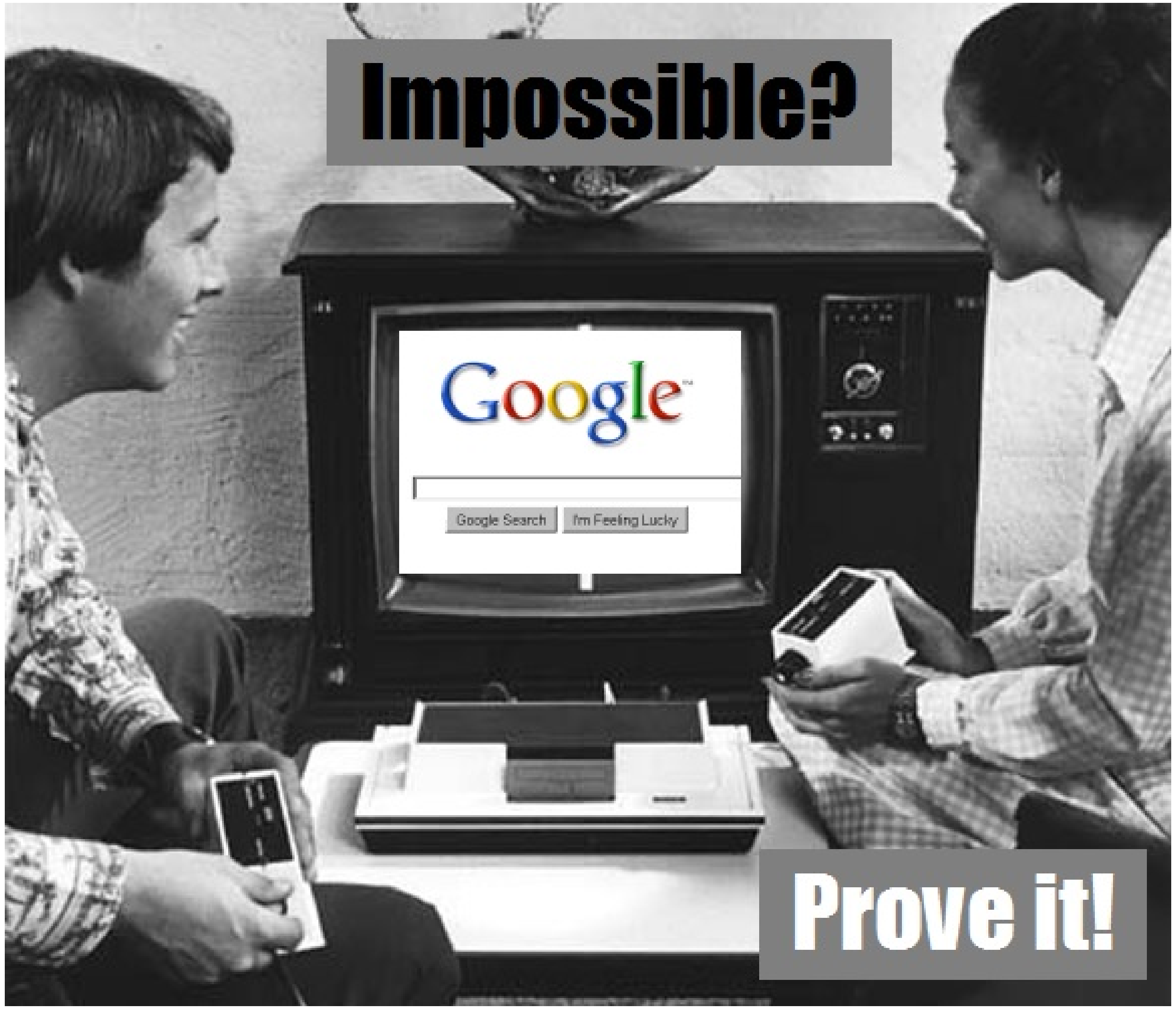
Why do lower bounds stop “just before” proving major results?

Emanuele Viola

Northeastern University

September 2019



3

2

1

Outline

- **History, conjectures, and upper bounds**
- Intermission: Natural proofs and fast crypto
- The lower bounds we have are best?
- Some recent connections and results

Can we multiply n-digit integers faster than n^2 ?

- Feeling: "As regards number systems and calculation techniques, it seems that the final and best solutions were found in science long ago"

- In 1950's, Kolmogorov conjectured time $\Omega(n^2)$
  Started a seminar with the goal of proving it

Can we multiply n-digit integers faster than n^2 ?

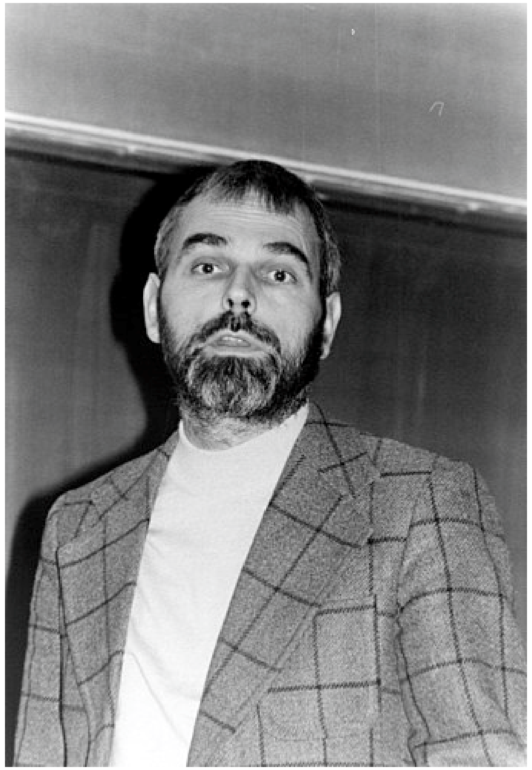
- Feeling: “As regards number systems and calculation techniques, it seems that the final and best solutions were found in science long ago”

- In 1950’s, Kolmogorov conjectured time $\Omega(n^2)$
  Started a seminar with the goal of proving it

- One week later, $O(n^{1.59})$ time by Karatsuba

- [..., 2007 Furer] $O(n \cdot log(n) \cdot \exp(\log^* n))$

Can we multiply nxn matrices faster than n^3 ?



1968 Strassen working to prove $\Omega(n^3)$

Can we multiply nxn matrices faster than n^3 ?

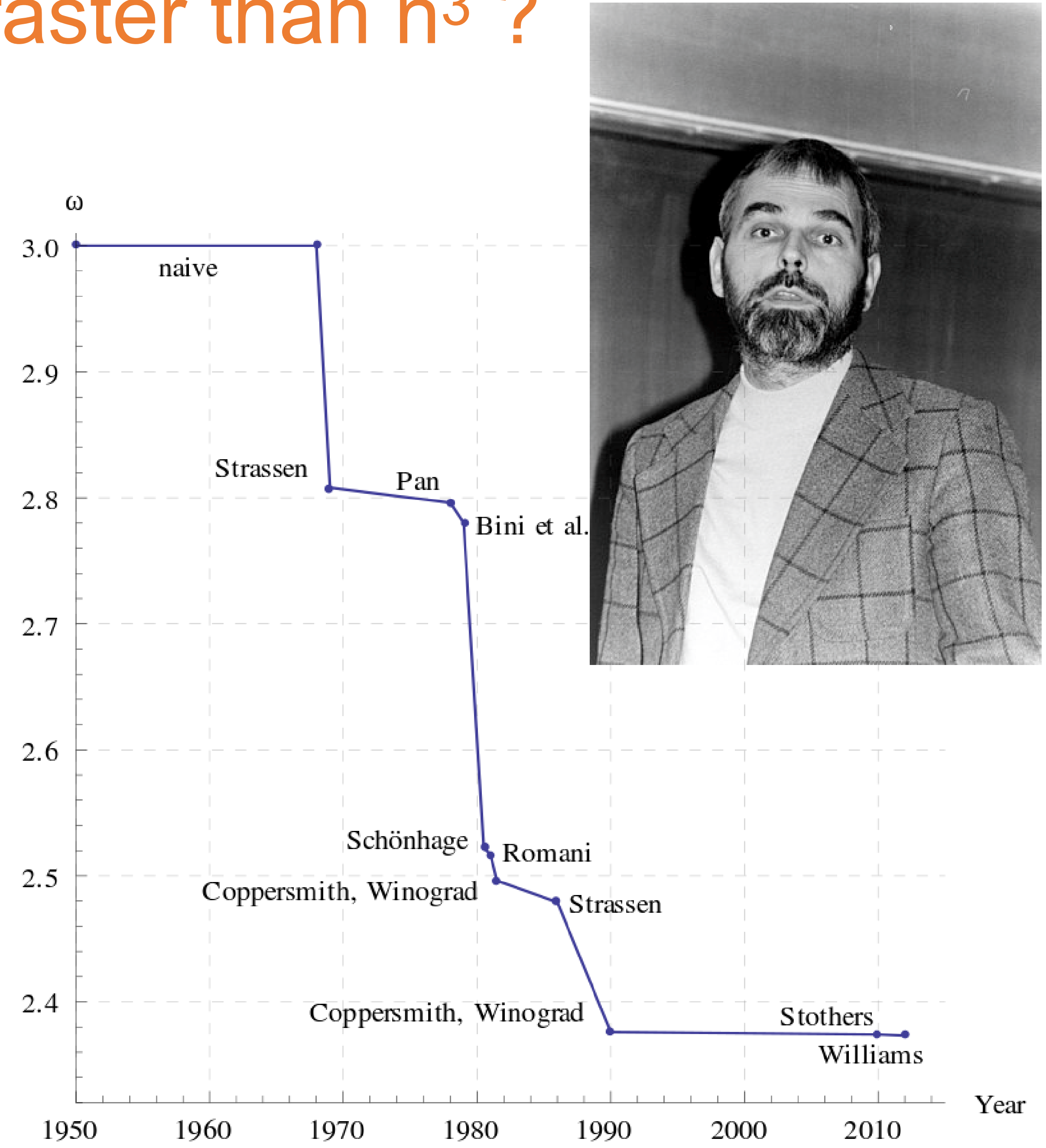
1968 Strassen working to prove $\Omega(n^3)$

1969: Volker Strassen.

Gaussian elimination is not optimal.

Numer. Math., 13:354–356, 1969.

$O(n^{2.81})$ algorithm



Proving lower bounds for linear transformations

Problem: Give explicit $n \times n$ matrix such that
linear transformation requires $\omega(n)$ size circuits

1970 Valiant:
Fourier transform matrix is a **super-concentrator**

Conjecture: Super-concentrators require $\omega(n)$ wires

Proving lower bounds for linear transformations

Problem: Give explicit $n \times n$ matrix such that

linear transformation requires $\omega(n)$ size circuits

1970 Valiant:

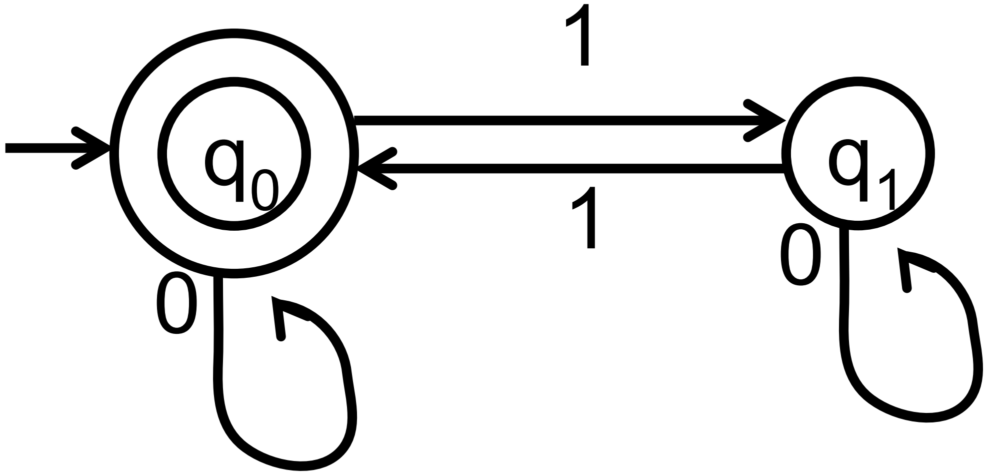
Fourier transform matrix is a **super-concentrator**

Conjecture: Super-concentrators require $\omega(n)$ wires

Later, Valiant: Super-concentrators with $O(n)$ wires exist

Space-bounded

Finite-state automata read input left to right



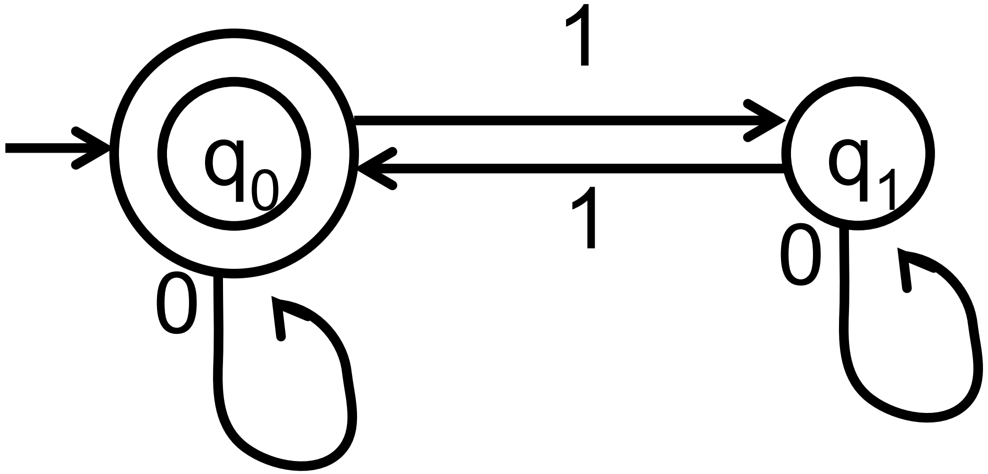
Theorem: Can't recognize palindromes

Let's allow them to read bits multiple times

Conjecture 1983 [Borodin, Dolev, Fich, Paul] Can't compute majority efficiently

Space-bounded

Finite-state automata read input left to right

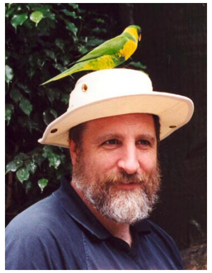


Theorem: Can't recognize palindromes

Let's allow them to read bits multiple times

Conjecture 1983 [Borodin, Dolev, Fich, Paul] Can't compute majority efficiently

Mix Barrington 1989: Can compute Majority (and NC^1)



Boolean circuits

Universal hash functions [Carter Wegman 79]

Conjecture 1990 [Mansour Nisan Tiwari]

Require super-linear size circuits

Boolean circuits

Universal hash functions [Carter Wegman 79]

Conjecture 1990 [Mansour Nisan Tiwari]

Require super-linear size circuits

Theorem 2008 [Ishai Kushilevitz Ostrovsky Sahai]

Linear-size suffices

Conjecture $P \neq NP$

Outline

- History, conjectures, and upper bounds

- **Intermission: Natural proofs and fast crypto**

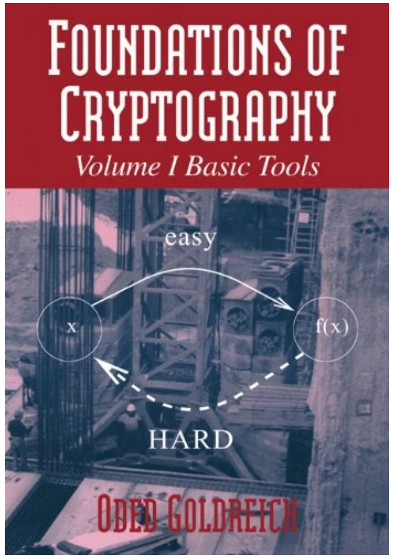
- The lower bounds we have are best?

- Some recent connections and results

Natural proofs [90's Razborov Rudich, Naor Reingold]

- If class C can compute **pseudorandom functions**,

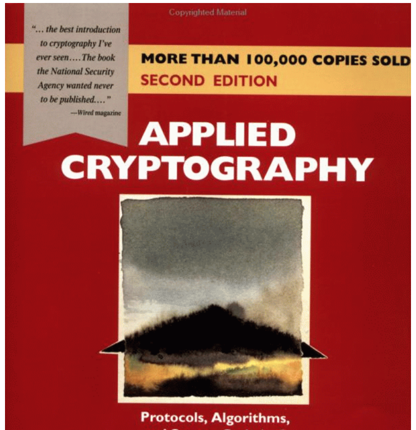
  Then proving lower bounds against C is "difficult"



- **theory** of cryptography

  Candidate pseudorandom functions in classes such as NC^1

  Somewhat far from state of lower bounds



- [Miles V] **practice** of cryptography

  Candidate more efficient pseudorandom functions

The SPN paradigm

[Shannon '49, Feistel-Notz-Smith '75]

S(ubstitution)-box

$S : GF(2^b) \longrightarrow GF(2^b)$

$x \mapsto x^{2^b-2}$

- computationally expensive
- “strong” crypto properties

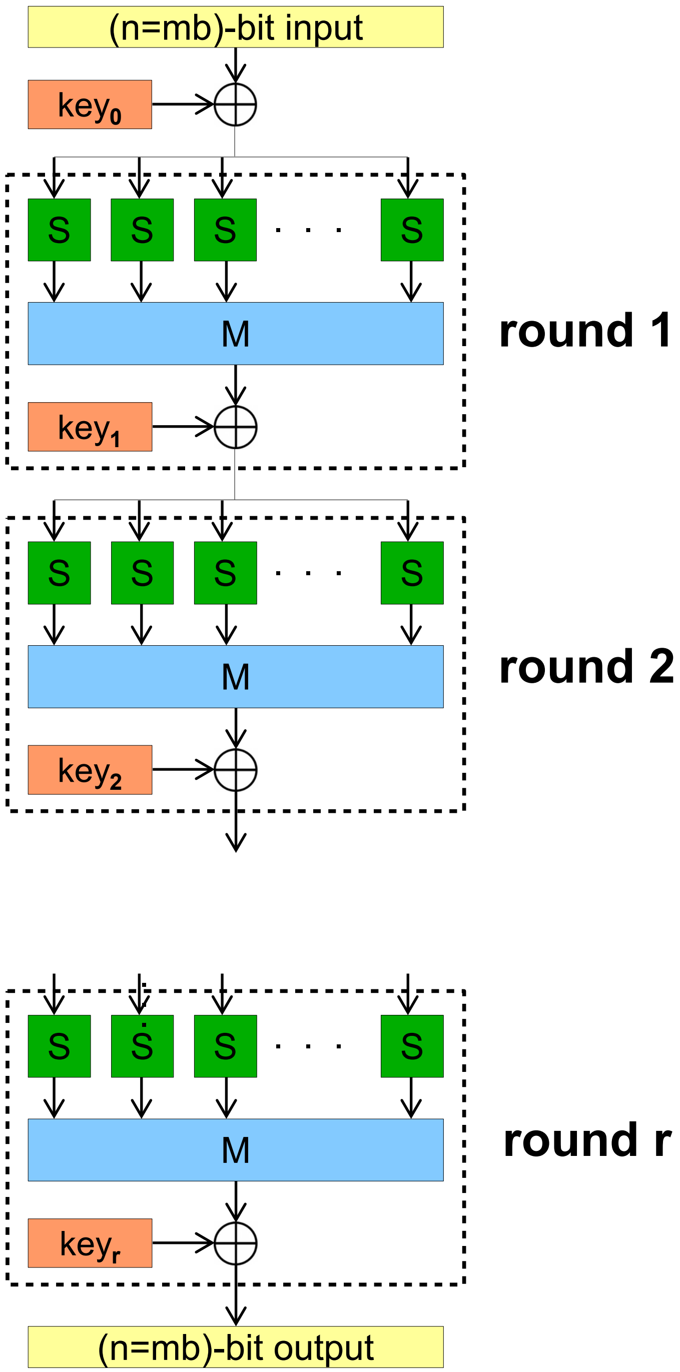
Linear transformation

$M : GF(2^b)^m \longrightarrow GF(2^b)^m$

- computationally cheap
- “weak” crypto properties

Key XOR

- only source of secrecy
- round keys = uniform, independent



[Miles V]

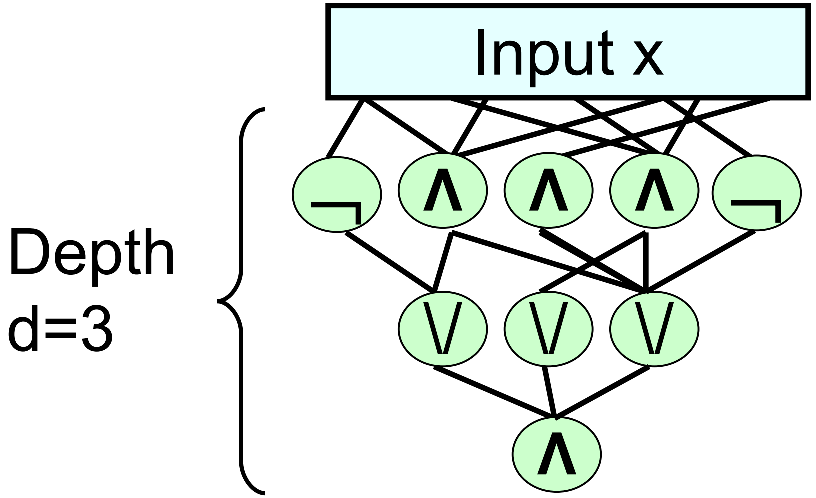
- Candidate pseudorandom function computable in quasi-linear time
- ... And in other models that will appear later in this talk
- Open: Construct more candidates from practical constructions

Outline

- History, conjectures, and upper bounds
- Intermission: Natural proofs and fast crypto
- **The lower bounds we have are best?**
- Some recent connections and results

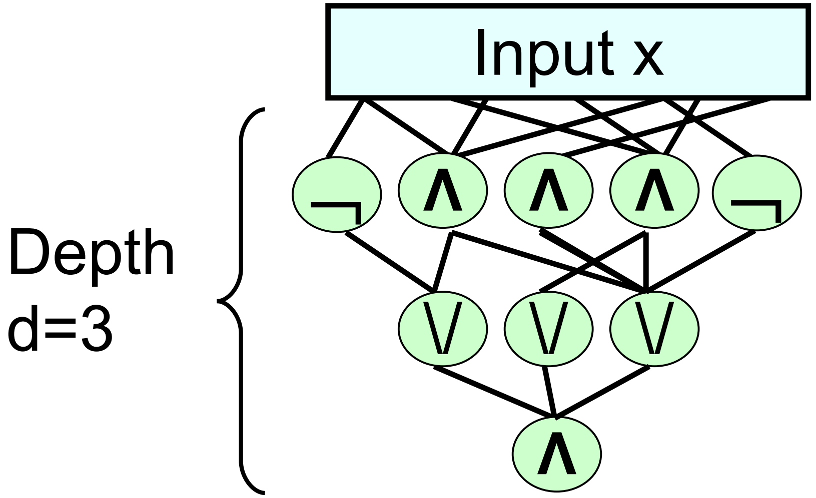
AC^0 circuits

- Depth-d, And-Or-Not circuits (AC^0)



- $2^{n^{\Omega(\frac{1}{d})}}$ lower bounds [80's: Furst Saxe Sipser, Ajtai, Yao, Hastad,…]
- Why not stronger bounds?

AC^0 circuits



- Depth-d, And-Or-Not circuits (AC^0)

- $2^{n^{\Omega(\frac{1}{d})}}$ lower bounds [80's: Furst Saxe Sipser, Ajtai, Yao, Hastad,…]

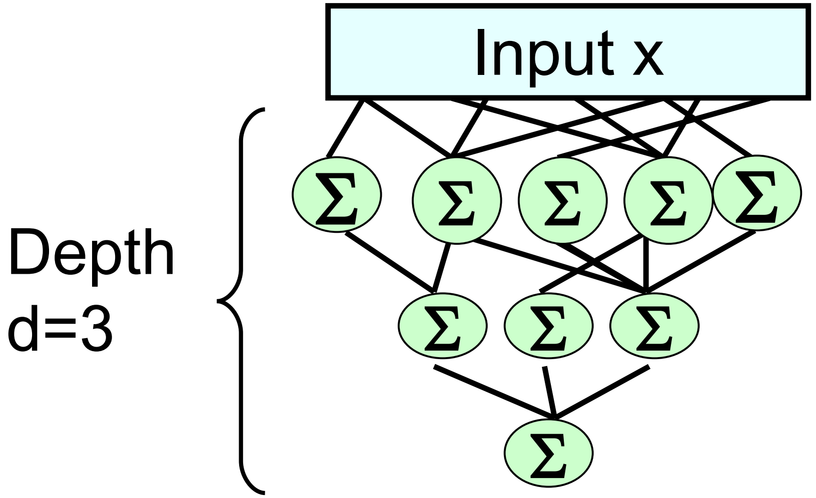
- Why not stronger bounds?

- Folklore: NC^1 has circuits of size $2^{n^{O(\frac{1}{d})}}$

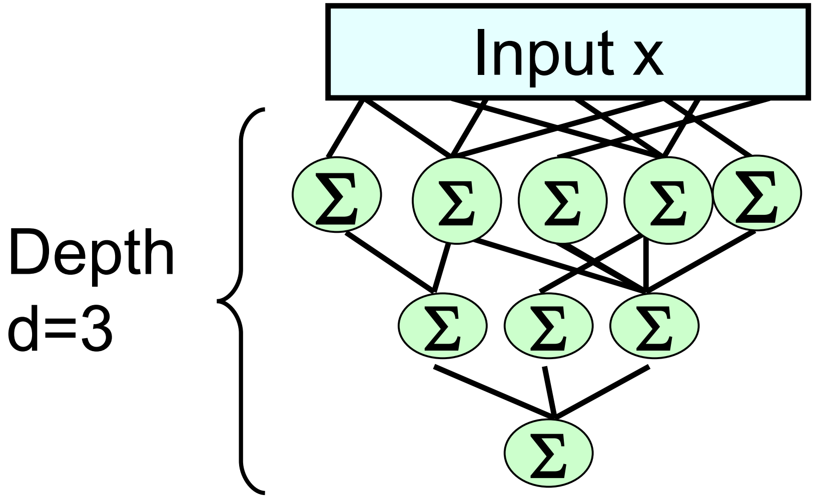
$\Rightarrow$ 80's bounds are best without proving major (false?) results

Threshold circuits

- f := product of n permutations on O(1) elements (NC^1 complete)
- [1997: Impagliazzo Paturi Saks] $n^{1+c^{-d}}$ lower bounds f

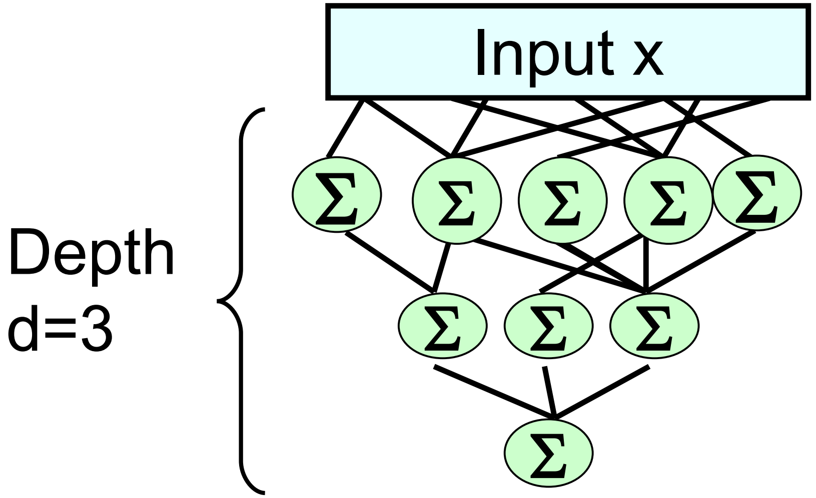


Threshold circuits



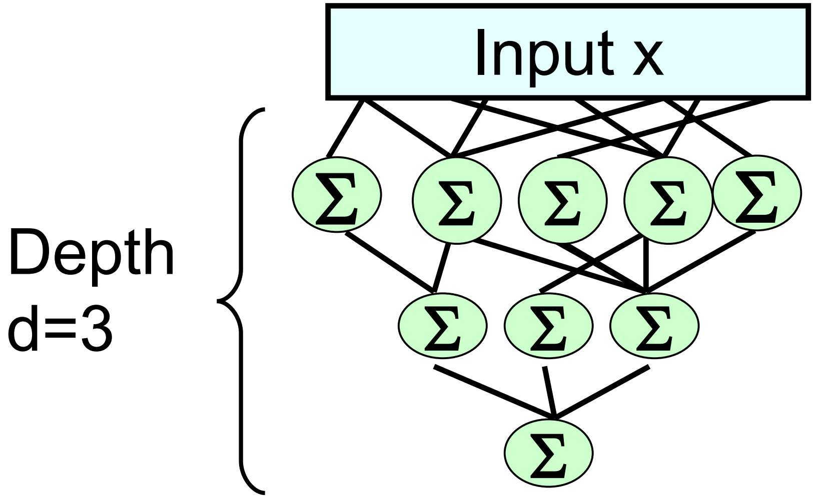
- f := product of n permutations
  on O(1) elements (NC^1 complete)
- [1997: Impagliazzo Paturi Saks] $n^{1+c^{-d}}$ lower bounds f
- [2010 Allender Koucky]: $NC^1 = TC^0 \Rightarrow f \text{ has size } n^{1+O(\frac{1}{d})}$

Threshold circuits



- f := product of n permutations on O(1) elements (NC^1 complete)
- [1997: Impagliazzo Paturi Saks] $n^{1+c^{-d}}$ lower bounds f
- [2010 Allender Koucky]: $NC^1 = TC^0 \Rightarrow f$ $has\ size\ n^{1+O(\frac{1}{d})}$
- [2015 Miles Viola]: $size\ n^{1+O(\frac{1}{d})}$ candidate pseudorandom function

Threshold circuits



- f := product of n permutations
  on O(1) elements (NC^1 complete)
- [1997: Impagliazzo Paturi Saks] $n^{1+c^{-d}}$ lower bounds f
- [2010 Allender Koucky]: $NC^1 = TC^0 \Rightarrow f\ has\ size\ n^{1+O(\frac{1}{d})}$
- [2015 Miles Viola]: $size\ n^{1+O(\frac{1}{d})}$ candidate pseudorandom function
- [2018 Chen Tell]: $NC^1 = TC^0 \Rightarrow f\ has$ size $n^{1+c^{-d}}$
  $\Rightarrow$ 1997 bound is best without proving major (false?) results

Proof [2018 Chen Tell]

- Recall: f = product of n permutations on O(1) elements (NC^1 complete)
- Theorem: $\exists\, k : f$ in size n^k & depth k $\Rightarrow \forall d : f$ in size $n^{1+c^{-d}}$ & depth O(d)

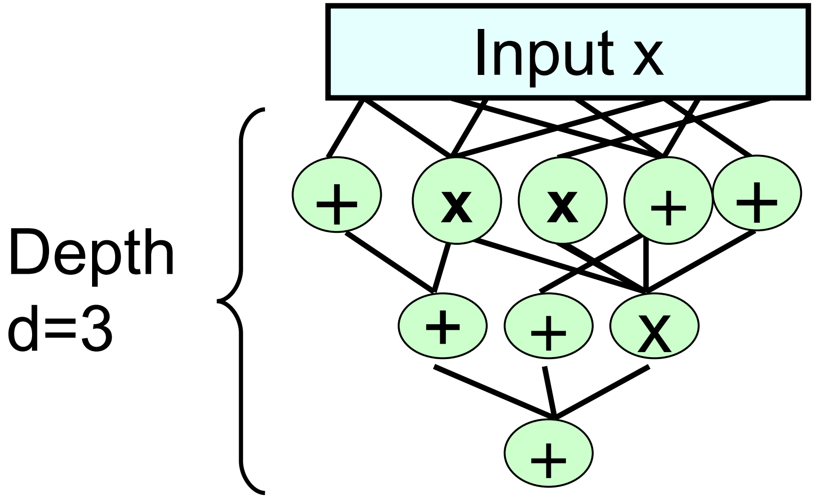
- Proof: Build a tree. Aim for size $n^{1+\epsilon}$
  $n_i :=$ number of nodes at level i (root level 0)

Level i fan-in: $(n^{1+\epsilon}/n_i)^{1/k}$ Recursion: $n_{i+1} = n_i \cdot (n^{1+\epsilon}/n_i)^{1/k}$

Solution: $n_i = n^{(1+\epsilon)(1-(1-1/k)^i)}$

Setting $i = O(k \log(1/\epsilon))$ gives $n_i > n$ QED

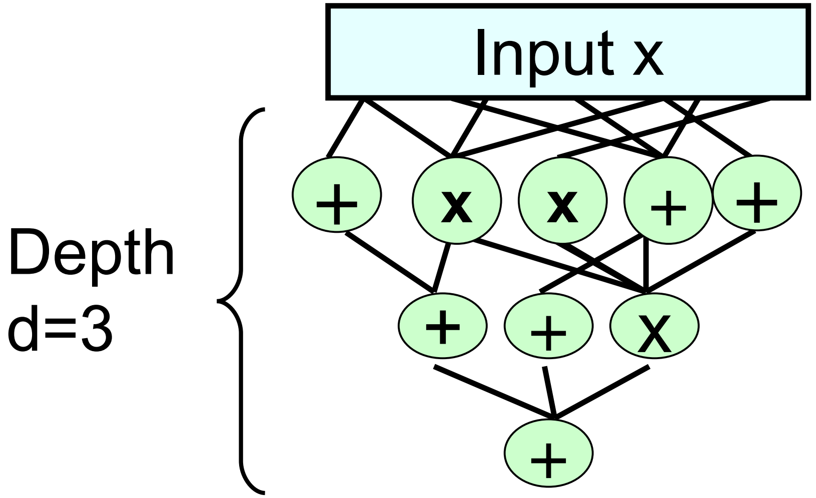
Algebraic complexity



- [2013 Gupta Kamath Kayal Saha Saptharishi]
  $n^{\Omega(\sqrt{n})}$ lower bounds for depth-4 homogeneous circuits

- Why not stronger bounds?

Algebraic complexity



- [2013 Gupta Kamath Kayal Saha Saptharishi]
  $n^{\Omega(\sqrt{n})}$ lower bounds for depth-4 homogeneous circuits

- Why not stronger bounds?

- [Agrawal Vinay, Koiran, Tavenas 2013]
  $n^{\omega(\sqrt{n})}$ lower bounds $\Rightarrow VP \neq VNP$

Why do current bounds stop “just before” proving major results?

Why do current bounds stop “just before” proving major results?

1. **No reason, it’s coincidence**
   I would find this “strange” because same bounds proved with seemingly different techniques

Why do current bounds stop “just before” proving major results?

1. **No reason, it’s coincidence**
   I would find this “strange” because same bounds proved with seemingly different techniques

2. **Current techniques are X, for major results need Y**

Why do current bounds stop “just before” proving major results?

1. **No reason, it’s coincidence**
   I would find this “strange” because same bounds proved with seemingly different techniques

2. **Current techniques are X, for major results need Y**

3. **Major results are false**

Outline

- History, conjectures, and upper bounds

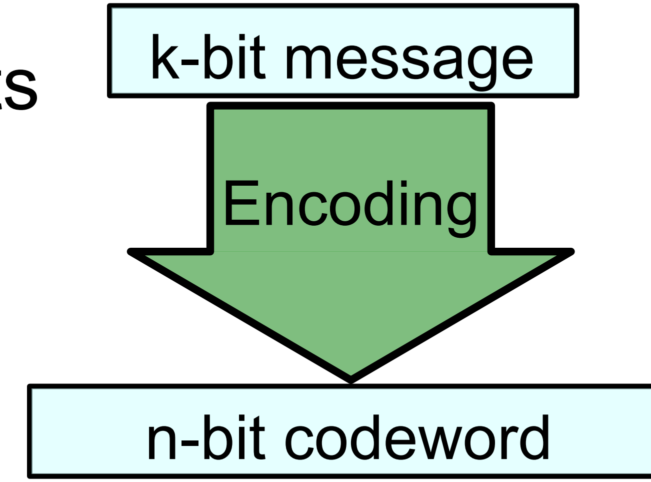
- Intermission: Natural proofs and fast crypto

- The lower bounds we have are best?

- Some recent connections and results
  - **Circuits encoding error-correcting codes**
  - Data structures
  - Turing machines

Complexity of error-correction encoding

- Asymptotically good code over {0,1}: $C \subseteq \{0,1\}^n$
  rate $\Omega(1)$: $|C| = 2^k$, $k = \Omega(n)$
  distance $\Omega(n)$: $\forall\ x \neq y \in C$, x and y differ in $\Omega(n)$ bits



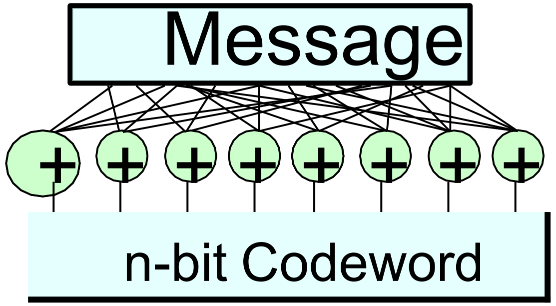
- Consider **encoding function** $f: \{0,1\}^k \to \{0,1\}^n$

- Want to compute f with circuits with **arbitrary** gates;
  only count number of wires

Previous work

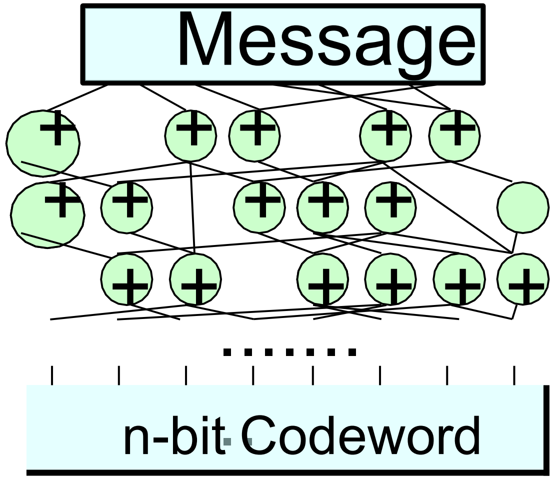
Depth 1   Wires $\Theta(n^2)$

Unbounded fan-in

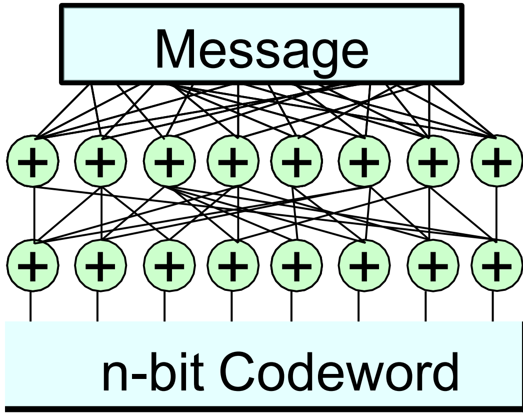


- Depth $O(\log n)$ Wires $\Theta(n)$

Fan-in 2
[Gelfand Dobrushin Pinsker 73]
[Spielman 95]

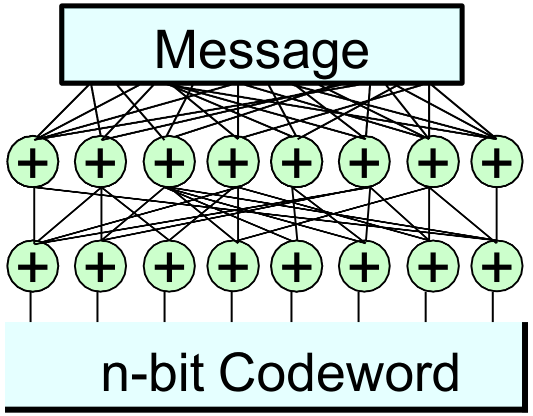


- Question: How many wires for depth 2?



[Gál Hansen Koucký Pudlák V 2012]

Depth	Wires
2	$n \cdot \Theta\left(\frac{\log n}{\log\log n}\right)^2$
d > 2	$n \cdot \Theta(\lambda_d(n))$

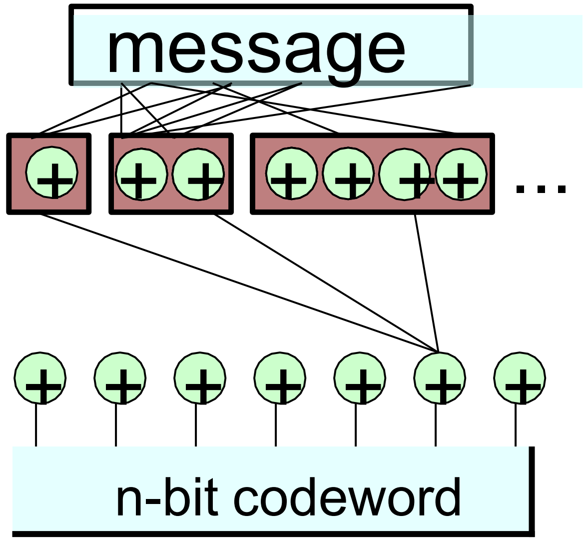


- λ inverse Ackermann: $\lambda_3(n)=\log\log n$, $\lambda_4(n)=\log^* n$, ...
- Best-known bound for linear function in NP

Probabilistic construction

Layer of log n blocks
∀ message ∃ balanced block

Output bit:
XOR one random bit per block



- i-th block balanced for message weight $w = \Theta(n/2^i)$
  Can do with wires $(n/w) \log \binom{n}{w} < n\,i$
- Total wires = $\Sigma_{i < \log n} (n\,i) + n \log n = n \cdot O(\log^2 n)$

Outline

- History, conjectures, and upper bounds

- Intermission: Natural proofs and fast crypto

- The lower bounds we have are best?

- Some recent connections and results
  - Circuits encoding error-correcting codes
  - **Data structures**
  - Turing machines

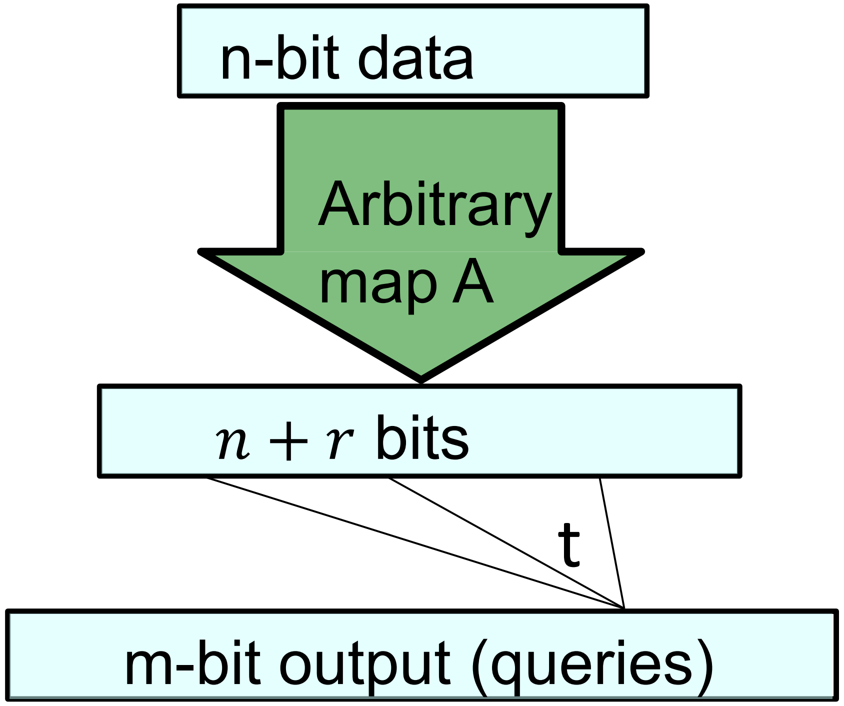
Static data structures

- Store n bits $x \in \{0,1\}^n$ into $n + r$ bits so that each of m queries can be answered reading t bits

- Trivial: $\mathrm{r} = m - n, t = 1 \; or \; r = 0, t = n$

- This talk: Think $r = o(n), m = O(n)$

- Best lower bound:
  $\mathrm{t} = \Omega\left(\frac{\mathrm{n}}{\mathrm{r}}\right)$ [‘07 Gal Miltersen]



From circuits to data structures [V 2018]

- Theorem:
  If $f : \{0,1\}^n \to \{0,1\}^m$ computable with w wires in depth d
  then f has data structure with space $n + r$ time $t = \left(\frac{w}{r}\right)^d$ for any r

- Corollaries:
  - f = encoding $\Rightarrow \mathrm{t} = O\left(\frac{\mathrm{n}}{\mathrm{r}}\right) \log^3 n$ [GHKPV], matches [Gal Miltersen] $\Omega\left(\frac{\mathrm{n}}{\mathrm{r}}\right)$
  - $\mathrm{t} > \left(\frac{\mathrm{n}}{\mathrm{r}}\right)^5$ for $f \in NP$ implies new circuit lower bounds

- Concurrent [Dvir Golovnev Weinstein]: broader regime, but linear model

From circuits to data structures [V 2018]

- Theorem:
  If $f: \{0,1\}^n \to \{0,1\}^m$ computable with w wires in depth d
  then f has data structure with space $n + r$ time $t = \left(\frac{w}{r}\right)^d$ for any r

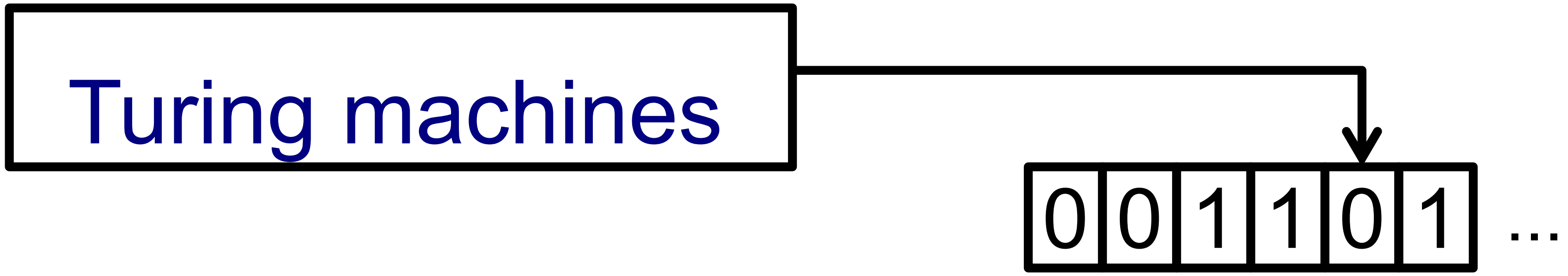
- Proof:
  Store n-bit input and values of gates with fan-in $> w/r$
  Number of such gates is $\leq r$
  To compute any gate: either you have it, or it depends on $\leq w/r$ gates at next layer, repeat. Qed

Open

- Data structures lower bounds for $\mathrm{r} = n^2, m = r^3$ imply anything?

Outline

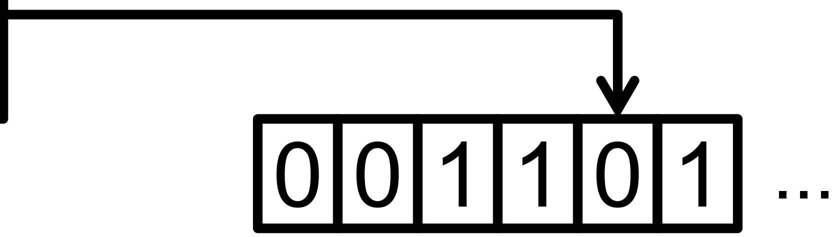
- History, conjectures, and upper bounds
- Intermission: Natural proofs and fast crypto
- The lower bounds we have are best?
- Some recent connections and results
  - Circuits encoding error-correcting codes
  - Data structures
  - **Turing machines**



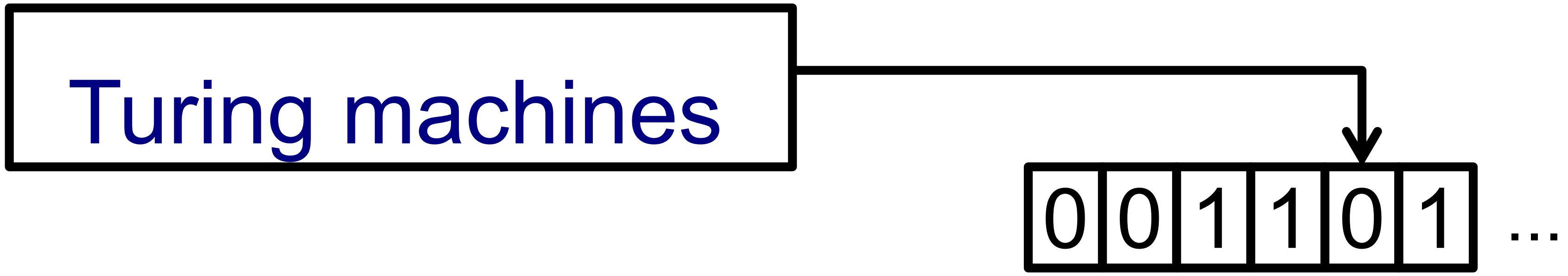
1) A useless model which only has historical significance

2) A fundamental challenge which lies right at the frontier of knowledge

Turing machines

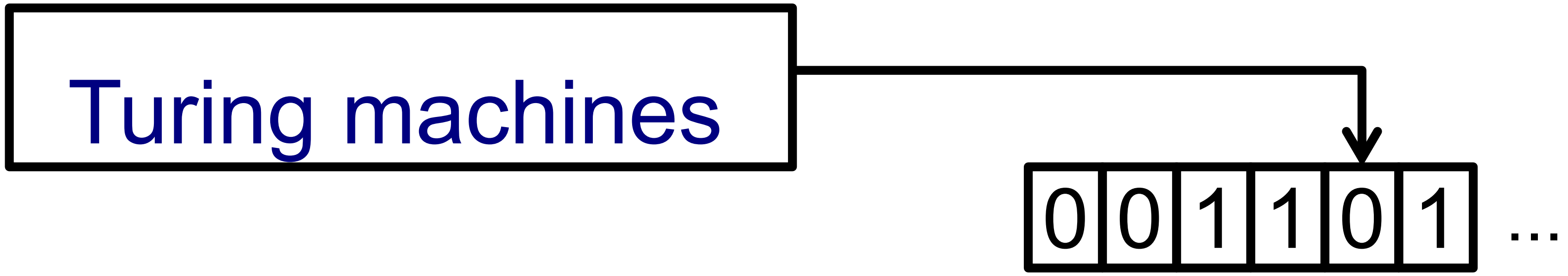


[Hennie 65] $\Omega(n^2)$ time lower bounds for 1-tape machines



[Hennie 65] $\Omega(n^2)$ time lower bounds for 1-tape machines

[Miles V] Candidate pseudorandom function in time $O(n^2)$

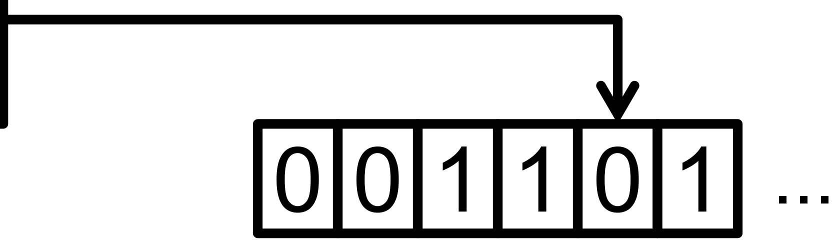


[Hennie 65] $\Omega(n^2)$ time lower bounds for 1-tape machines

[Miles V] Candidate pseudorandom function in time $O(n^2)$

Open: $n^{1+\Omega(1)}$ lower bounds for 2-tape machines

Turing machines



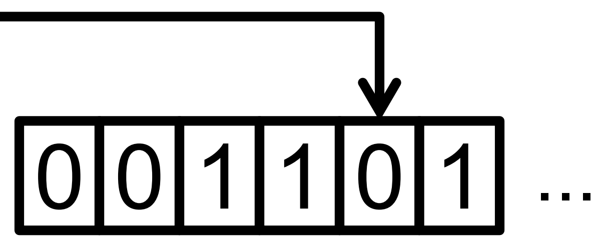
[Hennie 65] $\Omega(n^2)$ time lower bounds for 1-tape machines

[Miles V] Candidate pseudorandom function in time $O(n^2)$

Open: $n^{1+\Omega(1)}$ lower bounds for 2-tape machines

[Maass Schorr 87, van Melkebeek Raz, Williams] $n^{1+\Omega(1)}$ lower bounds for 2-tape machines but input tape read-only

Turing machines



[Hennie 65] $\Omega(n^2)$ time lower bounds for 1-tape machines

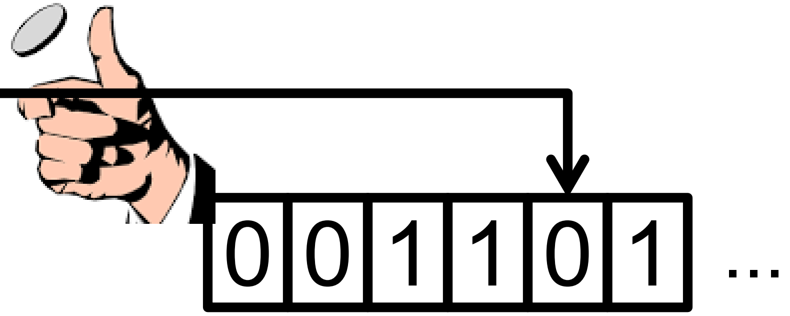
[Miles V] Candidate pseudorandom function in time $O(n^2)$

Open: $n^{1+\Omega(1)}$ lower bounds for 2-tape machines

[Maass Schorr 87, van Melkebeek Raz, Williams] $n^{1+\Omega(1)}$ lower bounds for 2-tape machines but input tape read-only

Question [V, Lipton, …]: What if the machine is randomized?

Turing machines



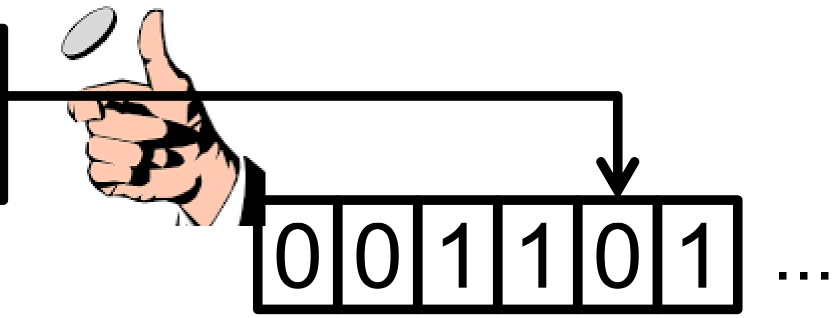
- [V 2019] $n^{1+\Omega(1)}$ lower bounds for 2-tape randomized machines but input tape read-only

- Key step of proof:
  Pseudorandom generator for 1-tape machines

- [1994 Impagliazzo Nisan Wigderson]
  Weaker model: Can fill the tape with bits that look random

- Need machine can toss coins at any point. This breaks

Turing machines



- First attempt to pseudorandom generator:
  bounded independence [Carter Wegman]

- Does not work

- **Bounded independence** and
  flip each bit independently with probability 0.01 (And recurse)

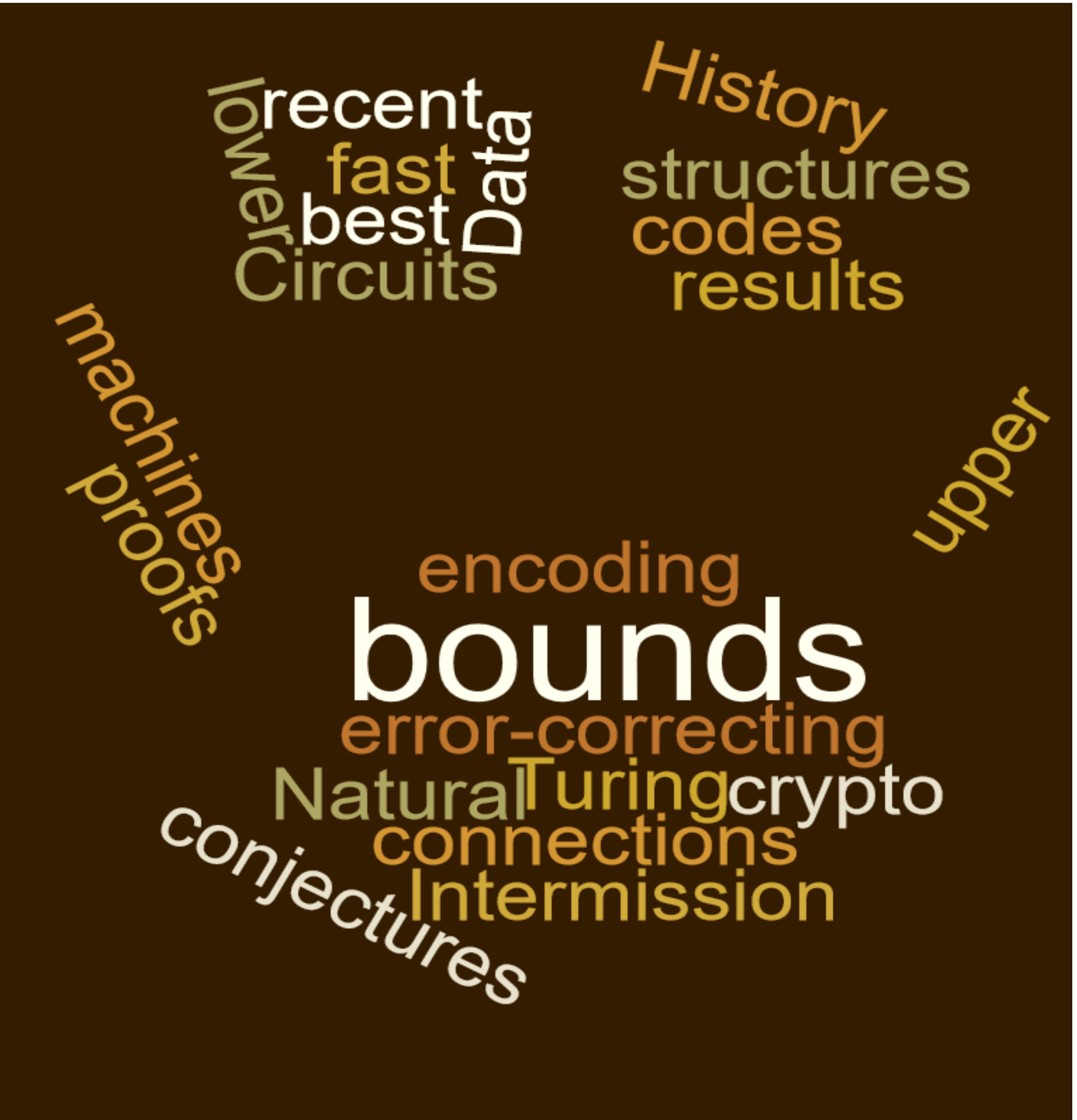
- Theorem: [Haramaty Lee V, …]
  Bounded independence plus noise fools small-space algorithms

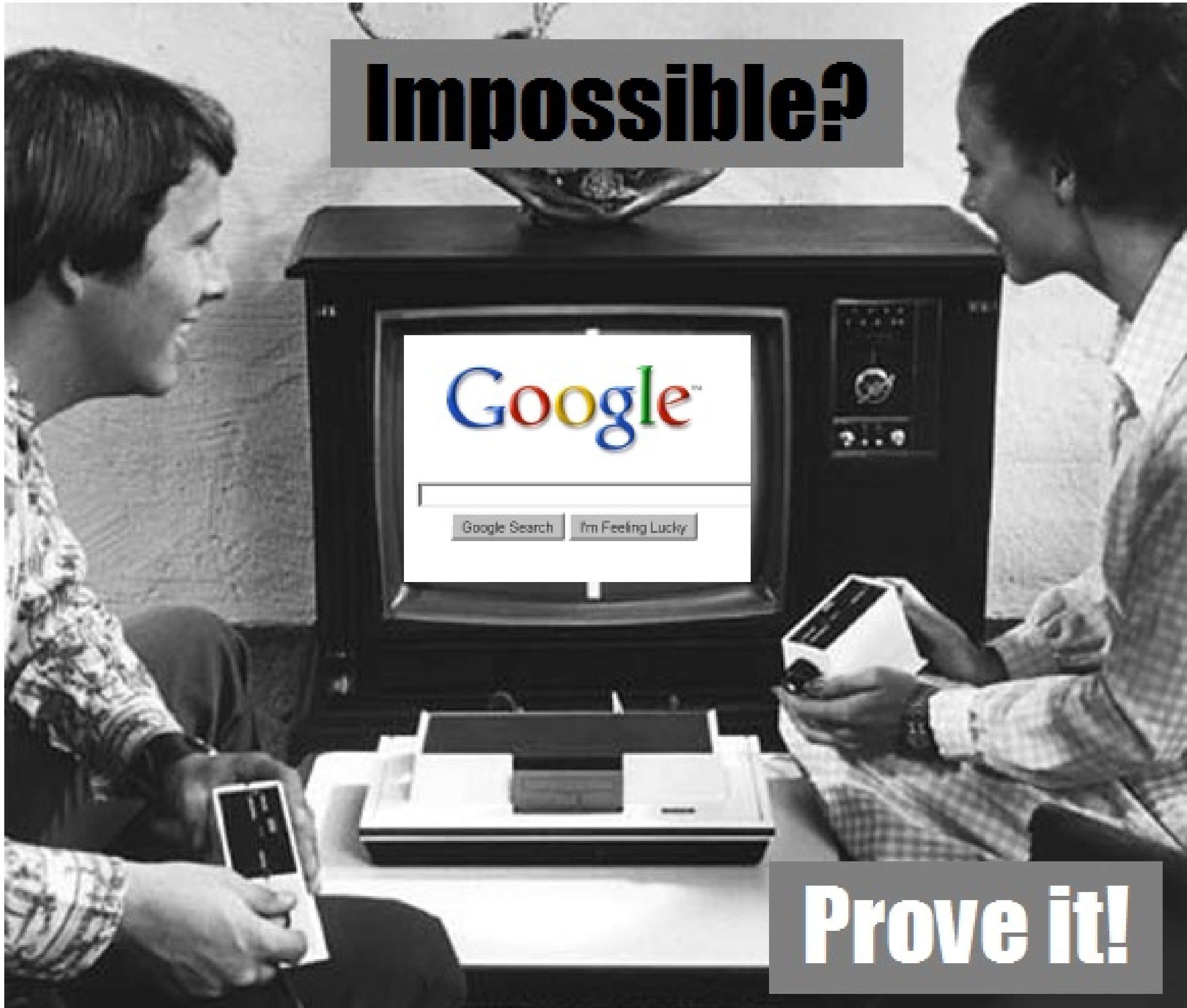
- Essentially simulate Turing machine computation with small space

Tape cell	1	2	3	4	5	6	7	8	9
	$\star$1								
		H							
			H						
				H					
					H				
						H			
							$\star$3		
							$\star$3		
						H			
							H		
						H			
					H				
				H					
			H						
		H							
			H						
				H					
			H						
		H							
			$\star$2						
				H					
					$\star$3				
						H			
							H		
								H	
							H		
								H	
							H		
								H	
									$\star$3
block	1	2	2	2	3	3	3	3	3
	b_1			b_2					b_3

Table 1: Computation table of an RTM with 9 work tape cells reading 6 random bits. Each row corresponds to a different time stamp and shows the position of the head H on the work tape. The symbol $\star$ indicates when a random bit is read. We have three boundaries shown at the bottom. The “block” row shows the partition of work cells in blocks. The induced partition on the random-bit tape is 133233.

Thanks!


lower
recent
fast
best
Data
Circuits
History
structures
codes
results
machines
proofs
upper
encoding
bounds
error-correcting
Natural
Turing
crypto
connections
Intermission
conjectures


Impossible?
Google
Google Search
I'm Feeling Lucky
Prove it!