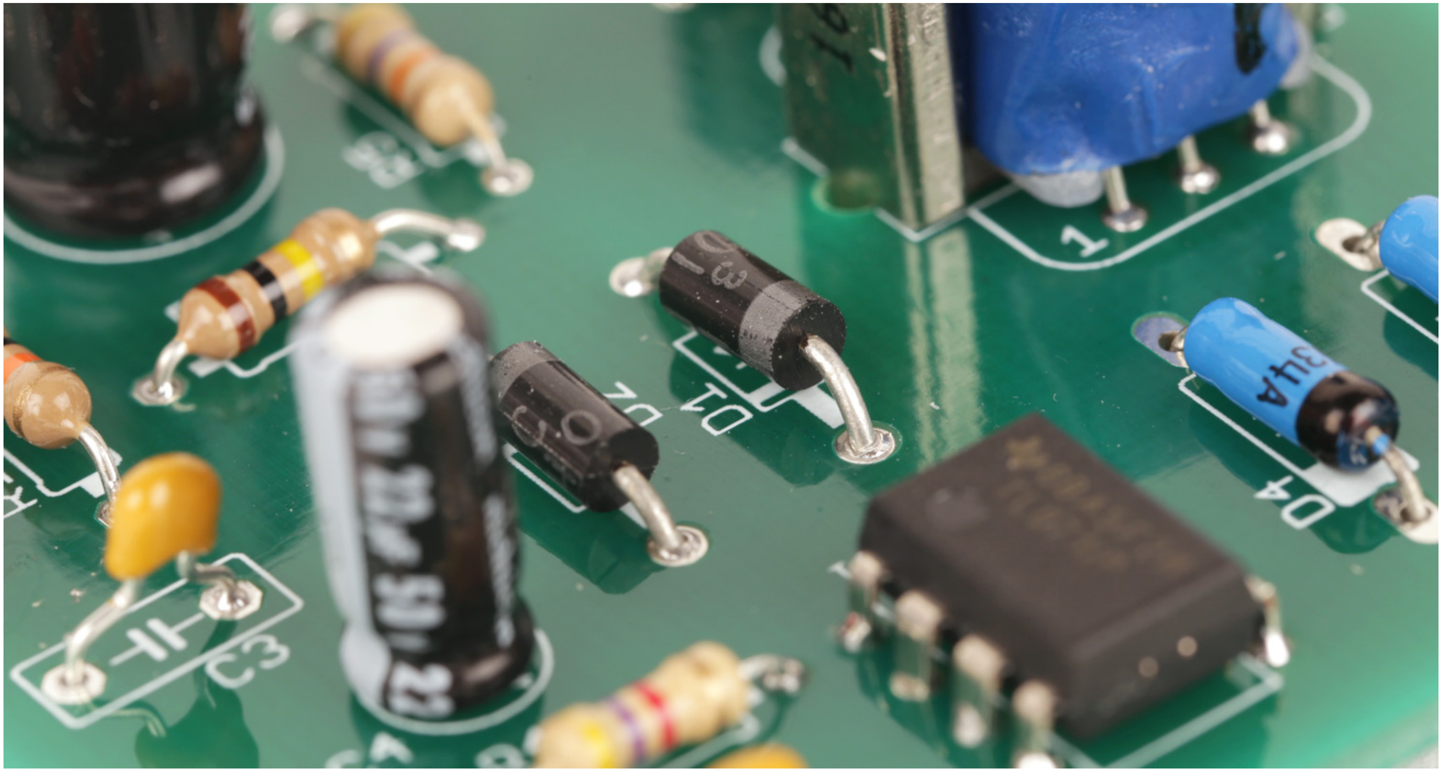
Gems of TCS

Circuit Complexity

Sasha Golovnev

March 18, 2021

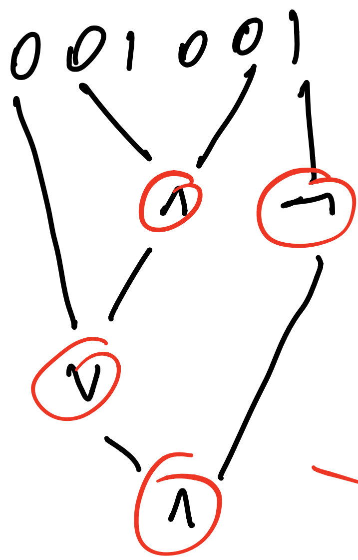

C3
D1
D4

P vs NP

For $P \neq NP$, for some NP-hard there is no poly-time algorithm

---

Circuits model



"time complexity"

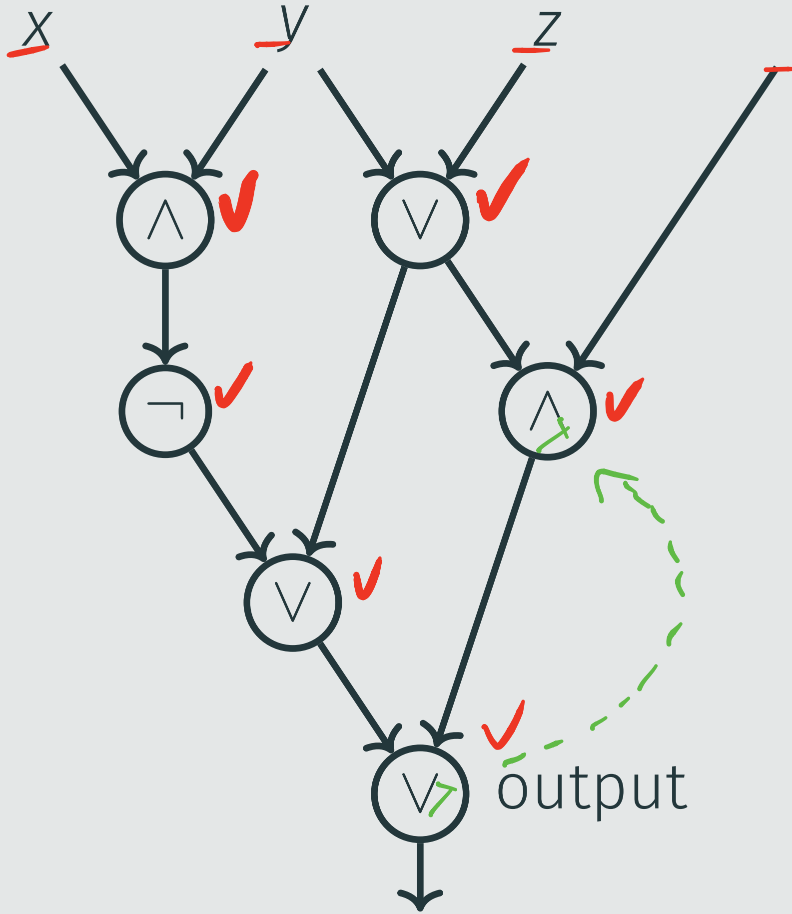
size of circuit:

= # gates,

don't count inputs

size of this circuit is 4

Circuit

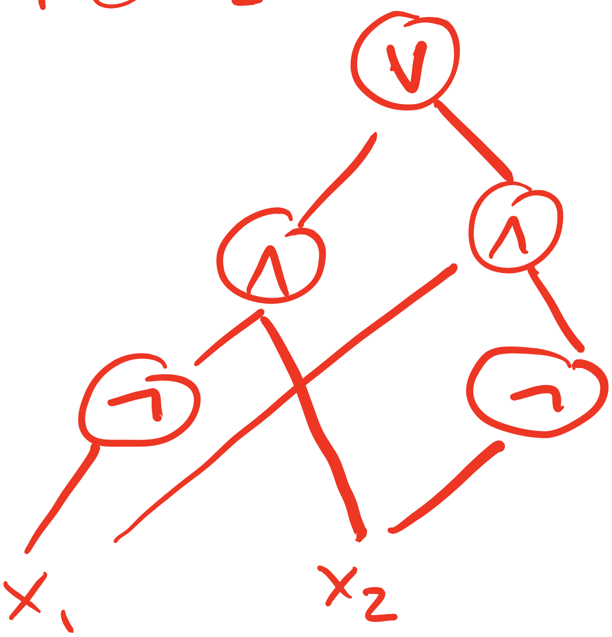


Definition

$x_1, \ldots, x_n \in \{0,1\}$

A circuit is a directed acyclic graph of in-degree at most 2. Nodes of in-degree 0 are called inputs and are marked by Boolean variables and constants. Nodes of in-degree 1 and 2 are called gates: gates of in-degree 1 are labeled with NOT, gates of in-degree 2 are labeled with AND or OR. One of the sinks is marked as output.

$x_1 \oplus x_2$



$\vee, \wedge, \neg$ to compute any function

$$x_1 \oplus x_2 \equiv \underline{(\neg x_1 \wedge x_2)} \vee \underline{(x_1 \wedge \neg x_2)}$$

x_1	x_2	$x_1 \oplus x_2$			
0	0	0	0	0	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

Boolean Circuits

$$f\colon \{0,1\}^n \to \{0,1\}$$

straight-line program

inputs: x_1, x_2, x_3

$$g_1 = \neg x_1$$
$$g_2 = x_2 \wedge x_3$$
$$g_3 = g_1 \vee g_2$$
$$g_4 = g_2 \vee 1$$
$$g_5 = g_3 \wedge g_4$$

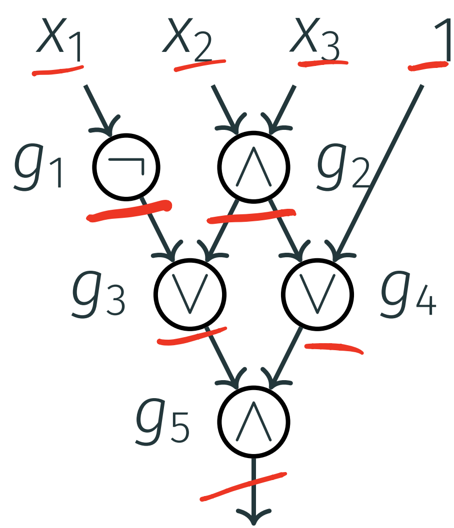
no branchings:
no if-statements
no loops

Boolean Circuits

$$f\colon \{0,1\}^n \to \{0,1\}$$

straight-line programs ≡ circuits

$$g_1 = \neg x_1$$
$$g_2 = x_2 \wedge x_3$$
$$g_3 = g_1 \vee g_2$$
$$g_4 = g_2 \vee 1$$
$$g_5 = g_3 \wedge g_4$$

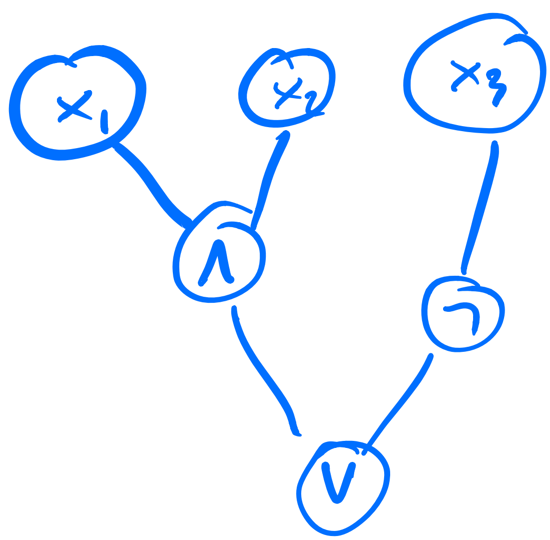


$f: \{0,1\}^n \to \{0,1\}$ - Boolean funcs

Algorithm for all values of n

For i from 1 to n:
    read (a[i])

Circuit for all values of n?



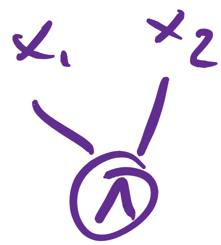
Circuits = Non-uniform algorithm

Circuit C solves $f:\{0,1\}^n \to \{0,1\}$ for every n,

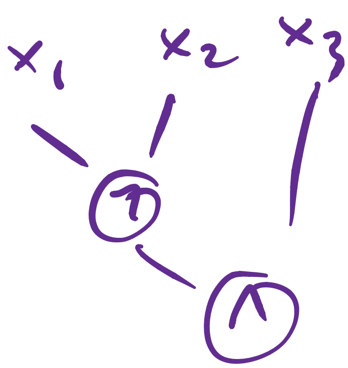
$C = C_1, C_2, C_3, C_4, \ldots\ldots$

C_1

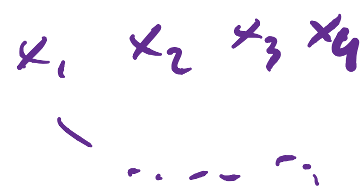
C_2



C_3



C_4



C solves f in size ("time") $\underline{10n+20}$, if size of each $C_i \le 10i+20$

Recall:

P – class of problems that can be solved in poly-time by algs

NP – class of problems whose solutions can be checked in poly-time

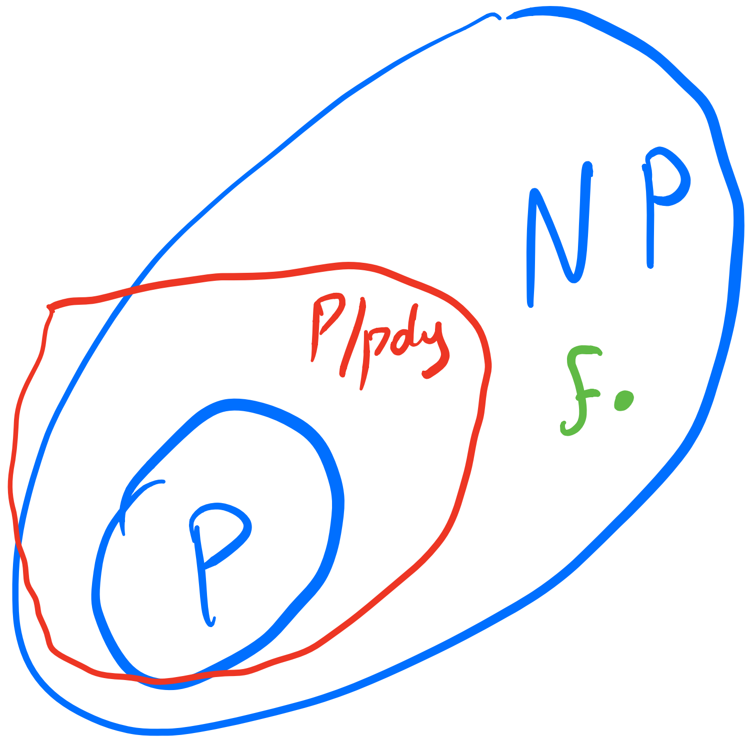
Def: $P/_{poly}$ – class of problems that can be solved by circuits of poly size

$$\boxed{P \subseteq P/poly}$$

Attack P vs NP question:

If find $f : \{0,1\}^n \to \{0,1\}$, $f \in NP$, $f \notin P/poly$ (cannot be solved by poly-size ckt)

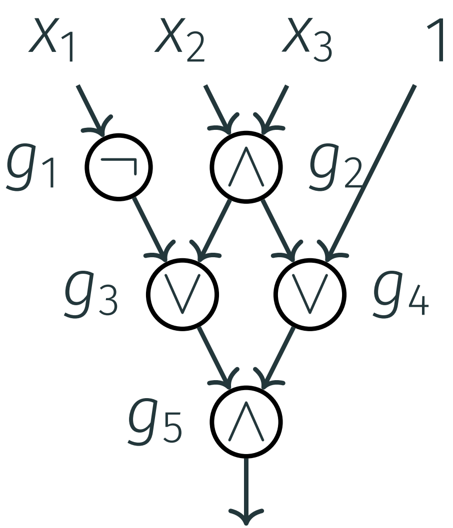
$\Longrightarrow \quad P \neq NP$


NP
P/poly
P
f.

Boolean Circuits

$$f\colon \{0,1\}^n \to \{0,1\}$$

$$\begin{aligned} g_1 &= \neg x_1 \\ g_2 &= x_2 \wedge x_3 \\ g_3 &= g_1 \vee g_2 \\ g_4 &= g_2 \vee 1 \\ g_5 &= g_3 \wedge g_4 \end{aligned}$$



Inputs:

$x_1, \ldots, x_n, 0, 1$

Gates:

AND, OR, NOT

Fan-out:

unbounded

Depth:

unbounded

Exponential Bounds

$f : \{0,1\}^n \to \{0,1\}$

Lower Bound [Sha1949]

Almost all functions of n variables have circuit size

$$\underline{\geq 2^n/n}$$

That is, almost all functions $\notin$ P/poly

For $P \neq NP$, we want $\boxed{f \in NP}$, that has complexity $\gg poly(n)$

Exponential Bounds

Lower Bound [Sha1949]

Almost all functions of n variables have circuit size

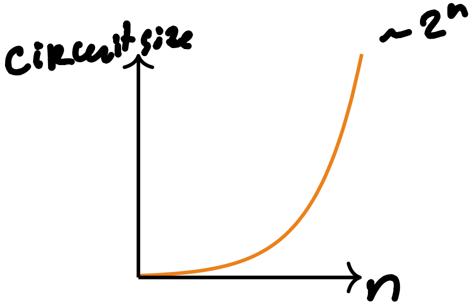
$$\geq 2^n/n$$

Upper Bound [Lup1958] $f : \{0,1\}^n \to \{0,1\}$

Any function can be computed by a circuit of size

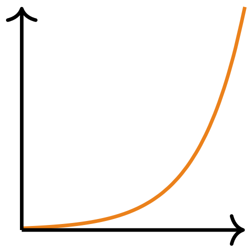
$$\leq \underline{2^n/n}$$

Explicit Bounds



Most functions have exponential circuit complexity

Explicit Bounds

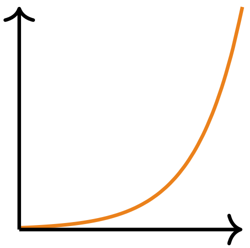


Most functions have exponential circuit complexity

$$P \neq NP$$

We want to prove super-polynomial lower bounds

Explicit Bounds

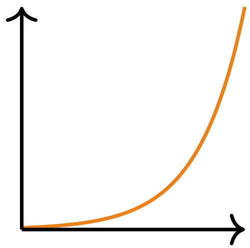


Most functions have exponential circuit complexity

$\mathsf{P} \neq \mathsf{NP}$

We want to prove super-polynomial lower bounds (for a function from **NP**)

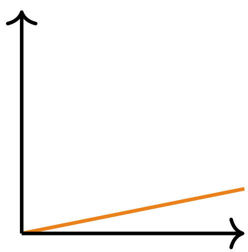
Explicit Bounds



Most functions have exponential circuit complexity

$P \neq NP$

We want to prove super-polynomial lower bounds (for a function from **NP**)

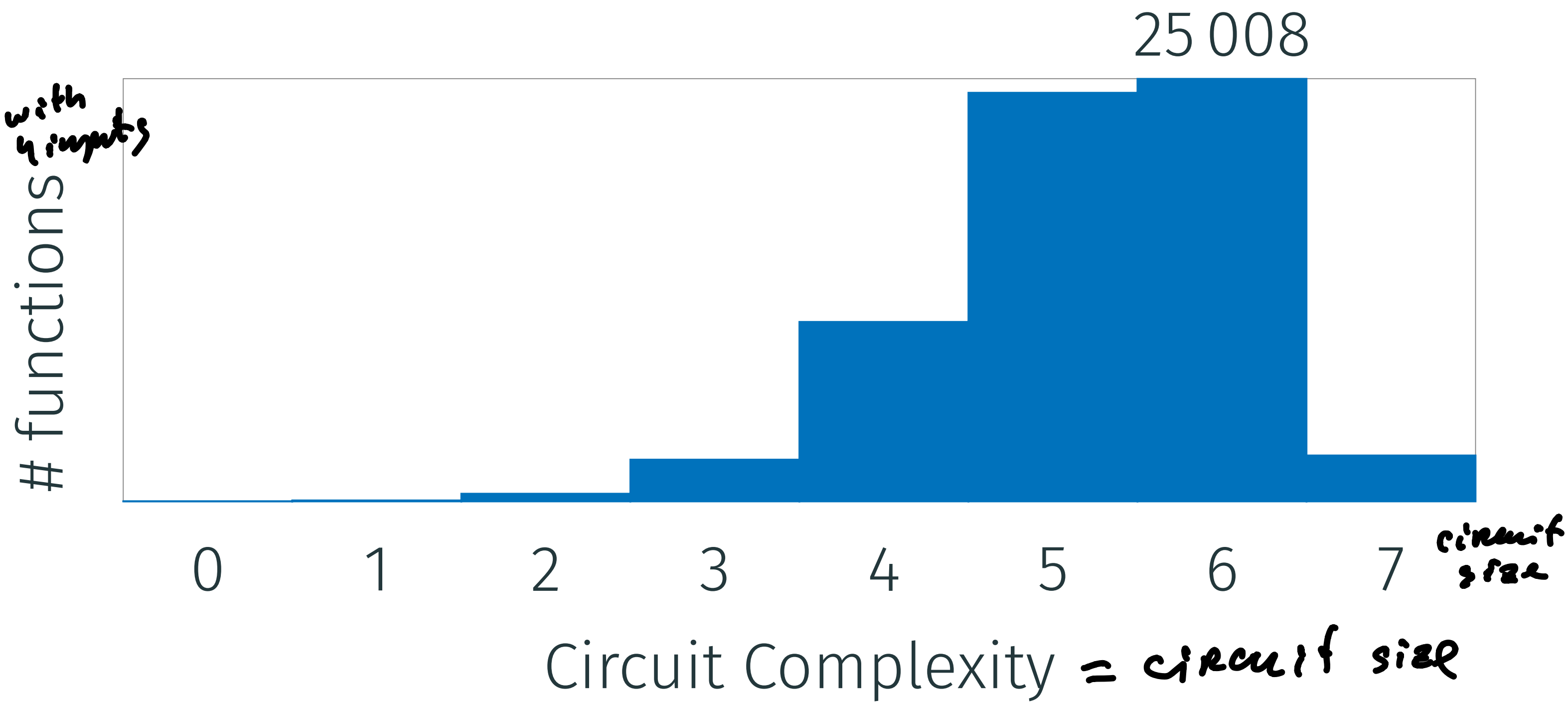


We can prove only $\approx 5n$ lower bounds

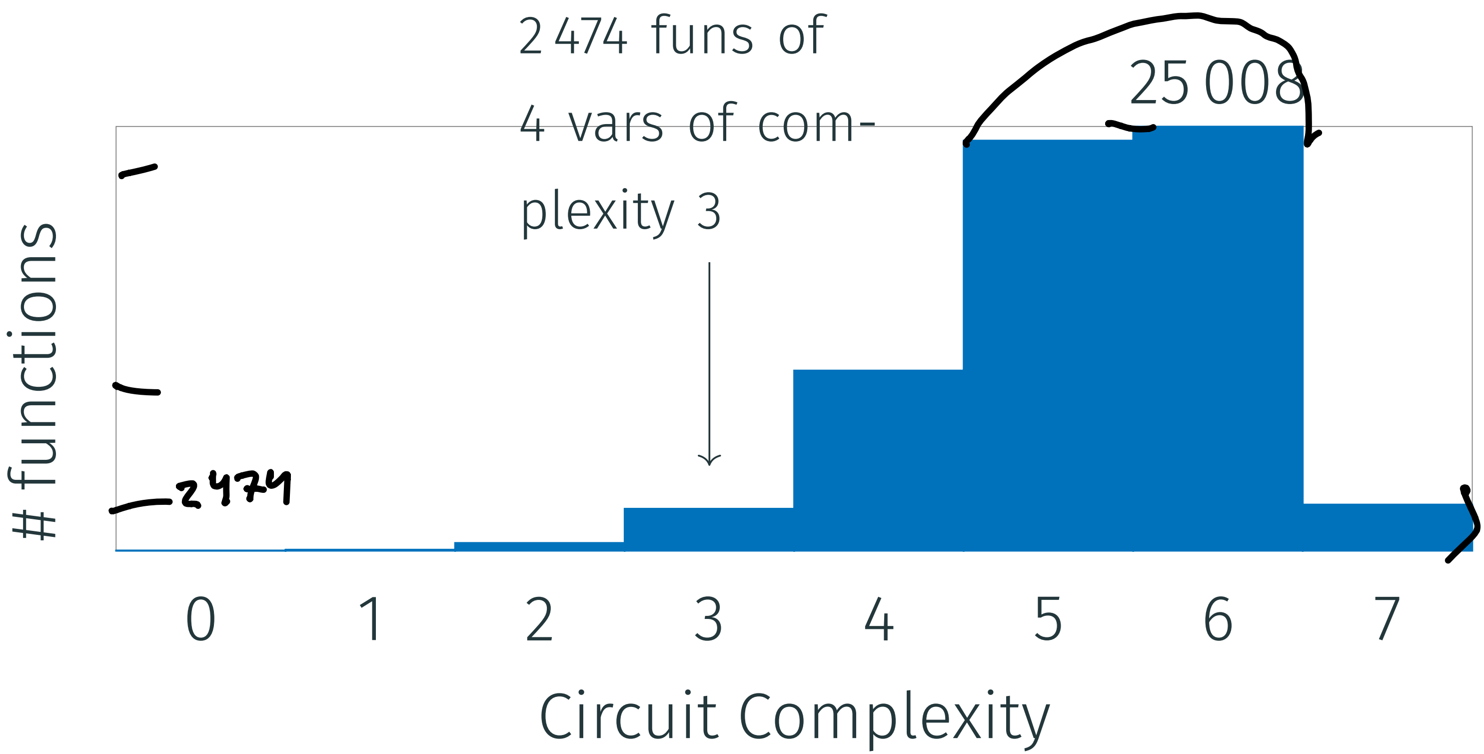
For a function from NP

Circuit Complexity: $n = 4$

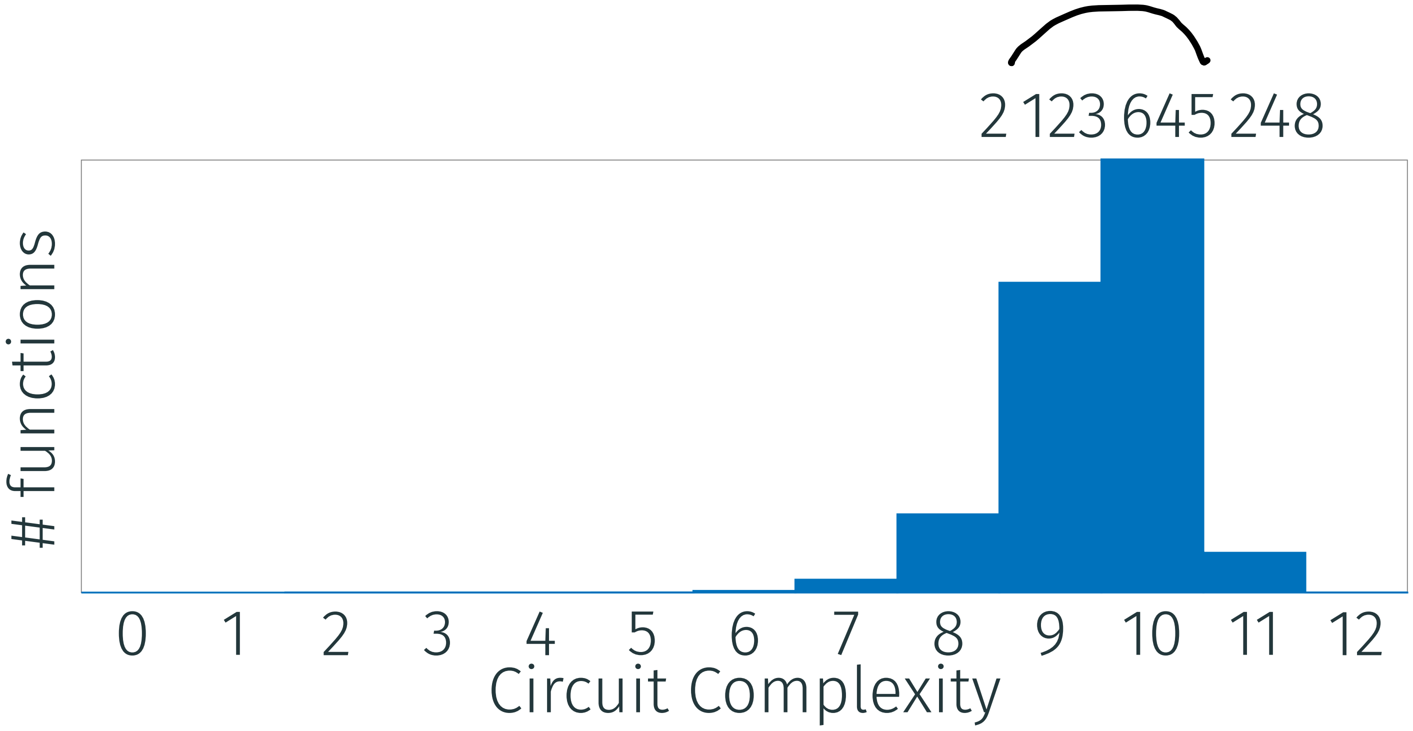
$f: \{0,1\}^4 \to \{0,1\}$



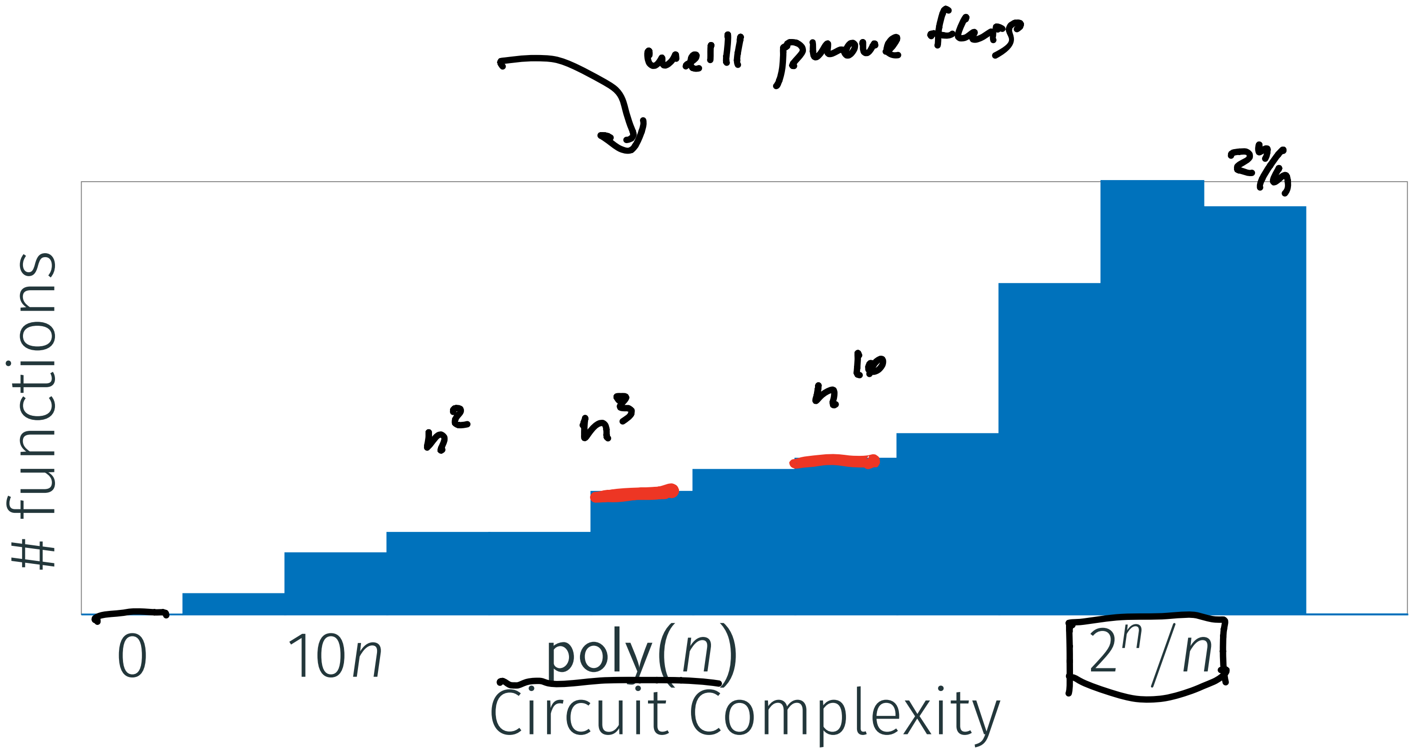
Circuit Complexity: $n = 4$



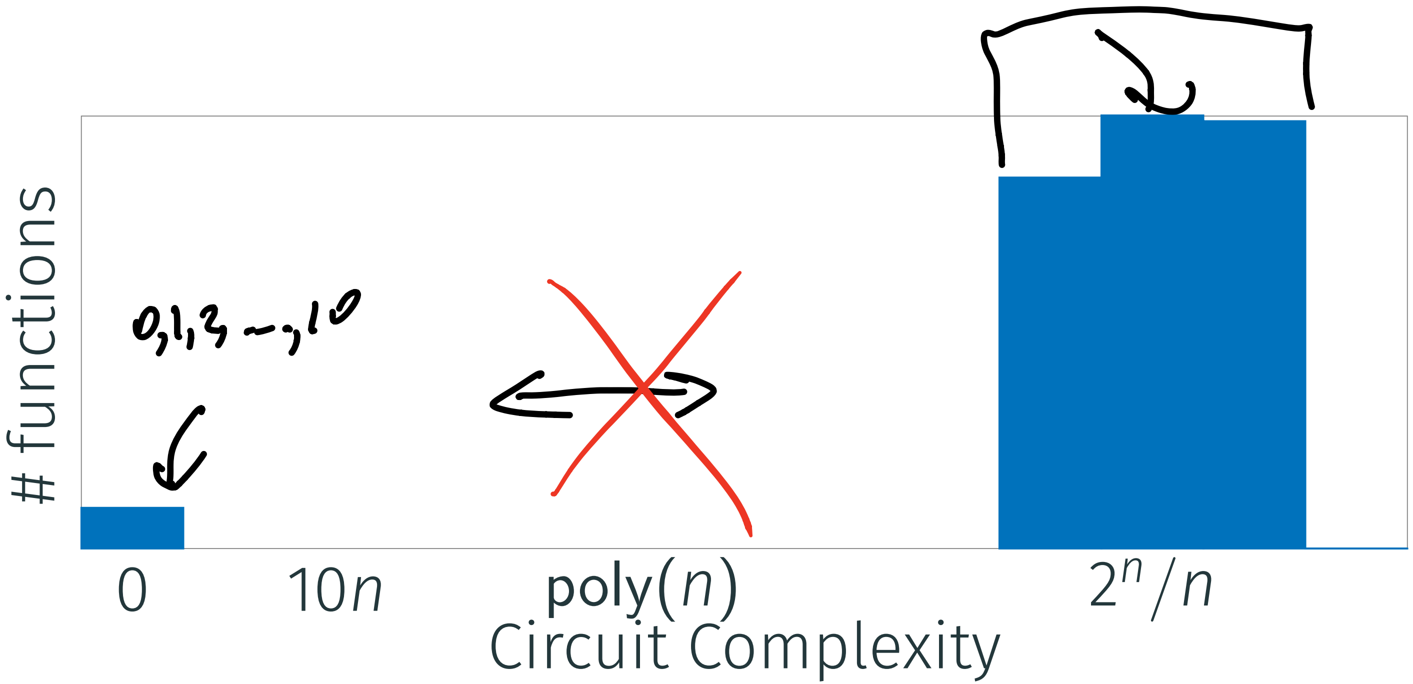
Circuit Complexity: $n = 5$



Circuit Complexity: General n



Circuit Complexity: General n



Hierarchy Theorem

Theorem

maximum circuit size

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

$T = n^2 \Rightarrow$ we'll find a problem whose complexity is $[n^2 - n, n^2 + n]$

$T = 1.5^n \Rightarrow$ $[1.5^n - n, 1.5^n + n]$

Hierarchy Theorem

Theorem

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

zero function

$$g_0(x) = 0\,, \forall x \in \{0,1\}^n$$

Hierarchy Theorem

Theorem

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

$g_0(x) = 0\,, \forall x \in \{0,1\}^n$

$\mathrm{Size}(g_0) = 1$

Hierarchy Theorem

Theorem

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

$\exists$ hard h

$g_0(x) = 0\,, \forall x \in \{0,1\}^n$ $\qquad$ $\mathrm{Size}(h) \geq 2^n/n$

$\mathrm{Size}(g_0) = 1$

Hierarchy Theorem

Theorem

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

$$g_0(x) = 0\,, \forall x \in \{0,1\}^n$$

$$\mathrm{Size}(g_0) = 1$$

$$\mathrm{Size}(h) \geq 2^n/n$$

$$h\colon \{0,1\}^n \to \{0,1\}$$

Hierarchy Theorem

Theorem

For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

zero fn

$$g_0(x) = 0\,, \forall x \in \{0,1\}^n$$

$$\mathrm{Size}(g_0) = 1$$

$$\mathrm{Size}(h) \geq 2^n/n$$

$$h : \{0,1\}^n \to \{0,1\}$$

$$\boxed{y_1}, \ldots, y_k \in \{0,1\}^n$$

$$h(y_i) = 1$$

otherwise
$h \to 0$

Hybrid Method

zero function

$g_0(x) = 1$ never

Hybrid Method

$g_0(x) = 1$ never

$g_1(x) = 1$ if $x = y_1$

Hybrid Method

$g_0(x) = 1$ never

$g_1(x) = 1$ if $x = y_1$

$g_2(x) = 1$ if $x \in \{y_1, y_2\}$

Hybrid Method

$g_0(x) = 1$ never

$g_1(x) = 1$ if $x = y_1$

$g_2(x) = 1$ if $x \in \{y_1, y_2\}$

$g_3(x) = 1$ if $x \in \{y_1, y_2, y_3\}$

Hybrid Method

$$g_0(x) = 1 \text{ never}$$

$$g_1(x) = 1 \text{ if } x = y_1$$

$$g_2(x) = 1 \text{ if } x \in \{y_1, y_2\}$$

$$g_3(x) = 1 \text{ if } x \in \{y_1, y_2, y_3\}$$

$$\ldots$$

$$g_k(x) = 1 \text{ if } x \in \underline{\{y_1, \ldots, y_k\}}$$

$g_k = h$ - hard function

Hybrid Method

$$g_0(x) = 1 \text{ never}$$

$$g_1(x) = 1 \text{ if } x = y_1$$

$$g_2(x) = 1 \text{ if } x \in \{y_1, y_2\}$$

$$g_3(x) = 1 \text{ if } x \in \{y_1, y_2, y_3\}$$

$$\cdots$$

$$h = g_k(x) = 1 \text{ if } x \in \{y_1, \ldots, y_k\}$$

Hybrid Method

$$g_0(x) = 1 \text{ never}$$

$$g_{i+1}(x) = g_i(x) \vee \boxed{(x = y_{i+1})}$$

$$g_1(x) = 1 \text{ if } x = y_1$$

$$g_2(x) = 1 \text{ if } x \in \{y_1, y_2\}$$

$$g_3(x) = 1 \text{ if } x \in \{y_1, y_2, y_3\}$$

$$\ldots$$

$$h = g_k(x) = 1 \text{ if } x \in \{y_1, \ldots, y_k\}$$

Hybrid Method

$$g_0(x) = 1 \text{ never}$$

$$g_1(x) = 1 \text{ if } x = y_1$$

$$g_2(x) = 1 \text{ if } x \in \{y_1, y_2\}$$

$$g_3(x) = 1 \text{ if } x \in \{y_1, y_2, y_3\}$$

$$\ldots$$

$$h = g_k(x) = 1 \text{ if } x \in \{y_1, \ldots, y_k\}$$

$$g_{i+1}(x) = g_i(x) \vee (x = y_{i+1})$$

If $y_{i+1} = 1011$

$$g_{i+1}(x) = \underline{g_i(x) \vee (x = 1011)}$$

Hybrid Method

$g_0(x) = 1$ never

$g_1(x) = 1$ if $x = y_1$

$g_2(x) = 1$ if $x \in \{y_1, y_2\}$

$g_3(x) = 1$ if $x \in \{y_1, y_2, y_3\}$

$\dots$

$h = g_k(x) = 1$ if $x \in \{y_1, \dots, y_k\}$

$$g_{i+1}(x) = g_i(x) \vee (x = y_{i+1})$$

$$g_{i+1}(x) = g_i(x) \vee (x = 1011)$$

$$g_{i+1}(x) = g_i(x) \boxed{\vee (x_1 \wedge \bar{x}_2 \wedge x_3 \wedge x_4)}$$

clause is 1 <=>
x = 1011

Hybrid Method

zero function
size = 1

$g_0(x) = 1$ never

$g_1(x) = 1$ if $x = y_1$

$g_2(x) = 1$ if $x \in \{y_1, y_2\}$

$g_3(x) = 1$ if $x \in \{y_1, y_2, y_3\}$

$\dots$

$h = g_k(x) = 1$ if $x \in \{y_1, \dots, y_k\}$

hard function
circuit size $\approx 2^n/n$

$$g_{i+1}(x) = g_i(x) \vee (x = y_{i+1})$$

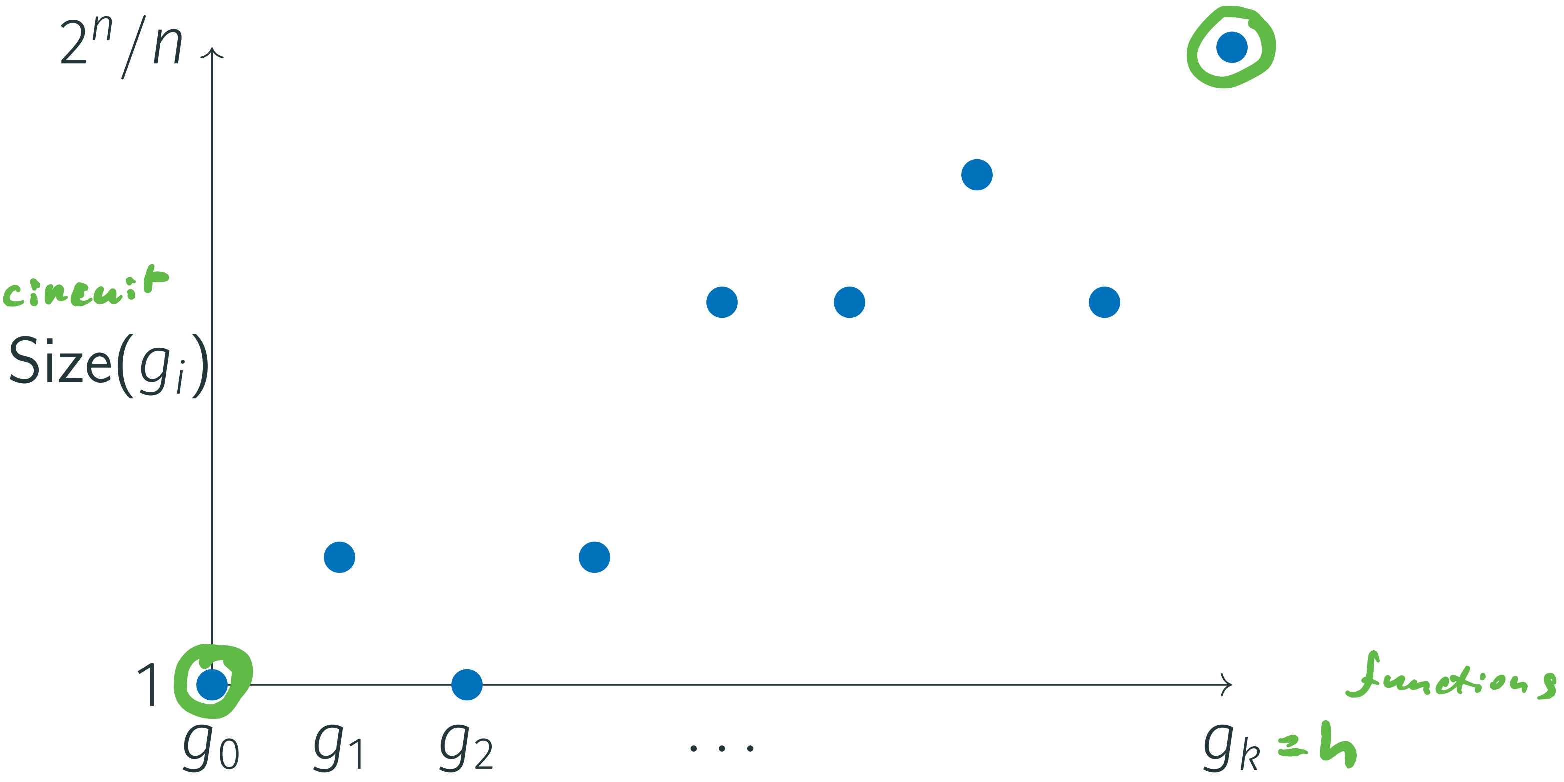
$$g_{i+1}(x) = g_i(x) \vee (x = 1011)$$

$$g_{i+1}(x) = g_i(x) \vee (x_1 \wedge \bar{x}_2 \wedge x_3 \wedge x_4)$$

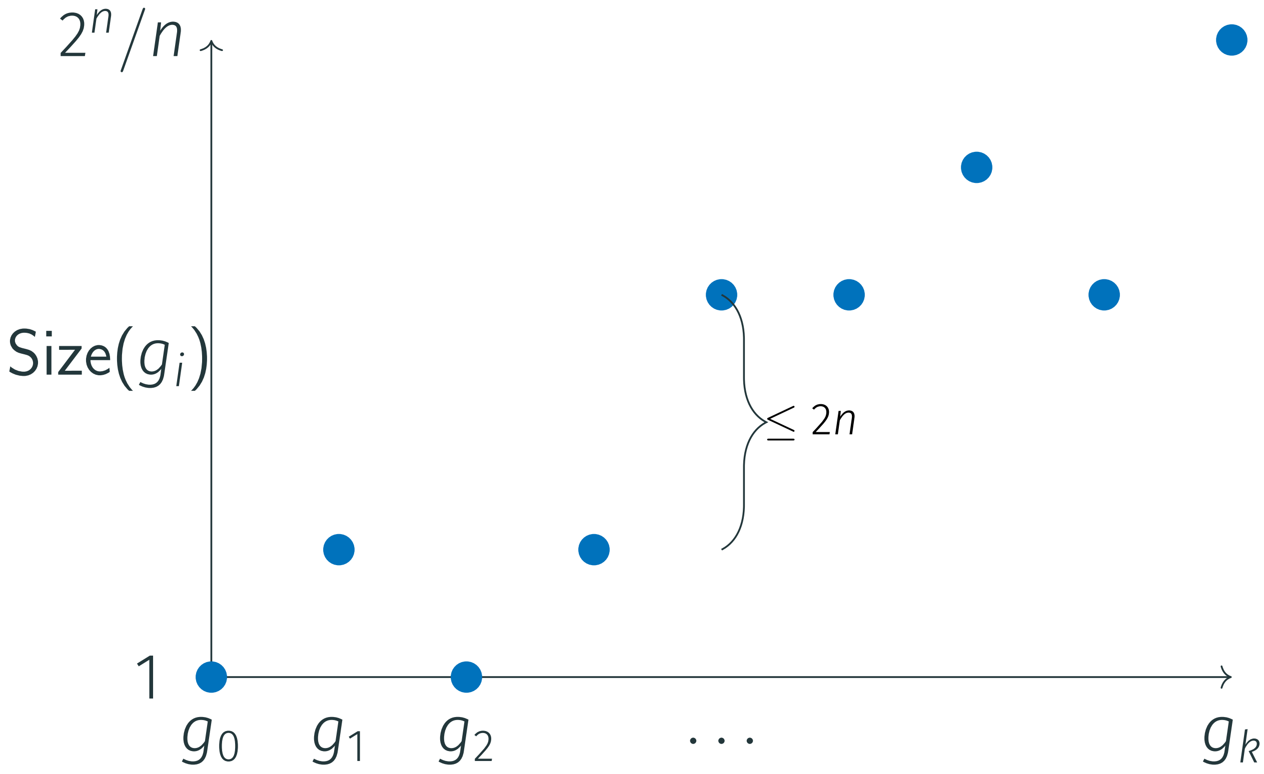
$$\text{Size}(g_{i+1}) \leq \text{Size}(g_i) + \boxed{2n}$$

$\text{Size}(g_3) \leq \text{Size}(g_2) + 2n$

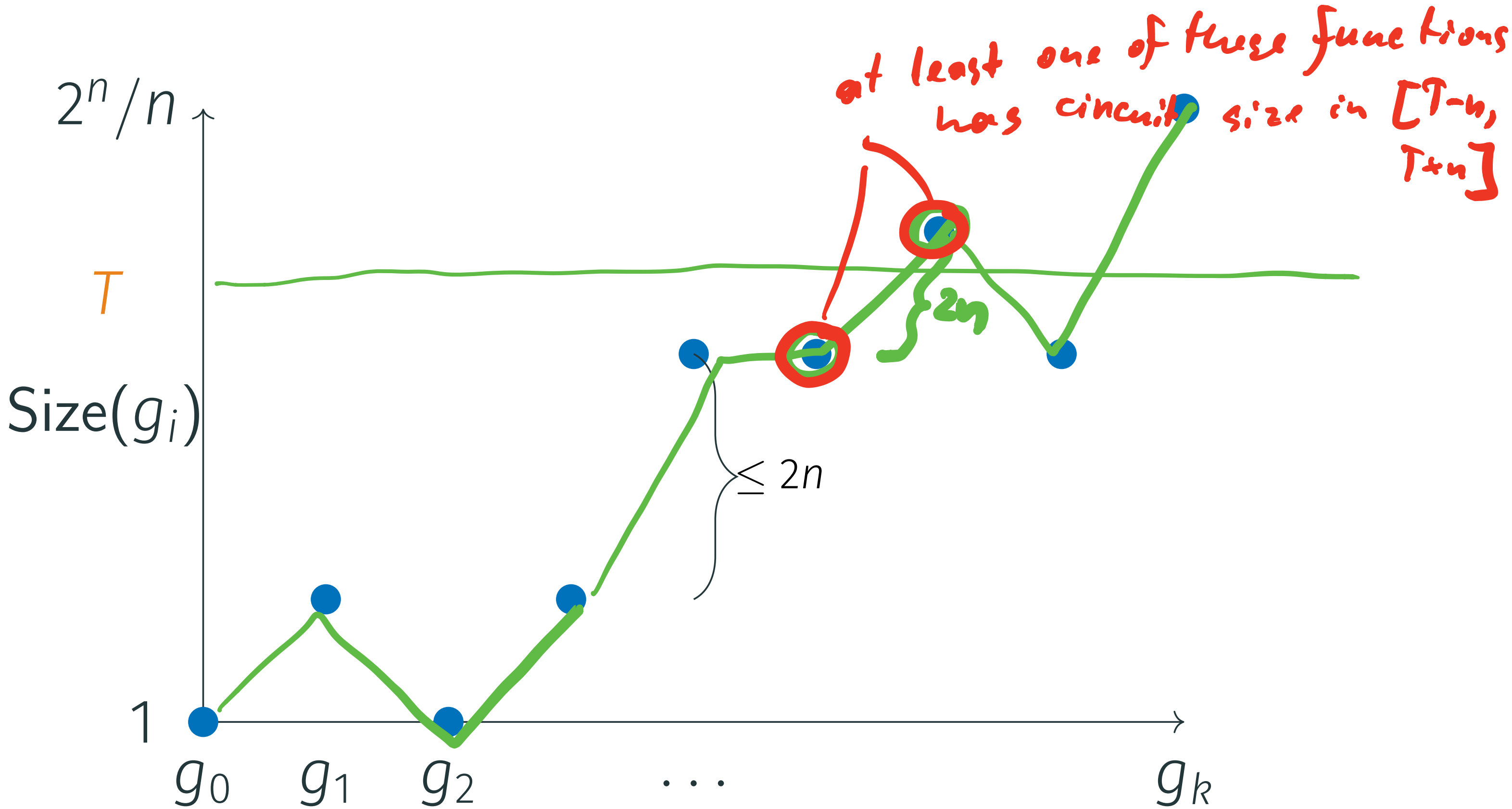
Hierarchy Theorem



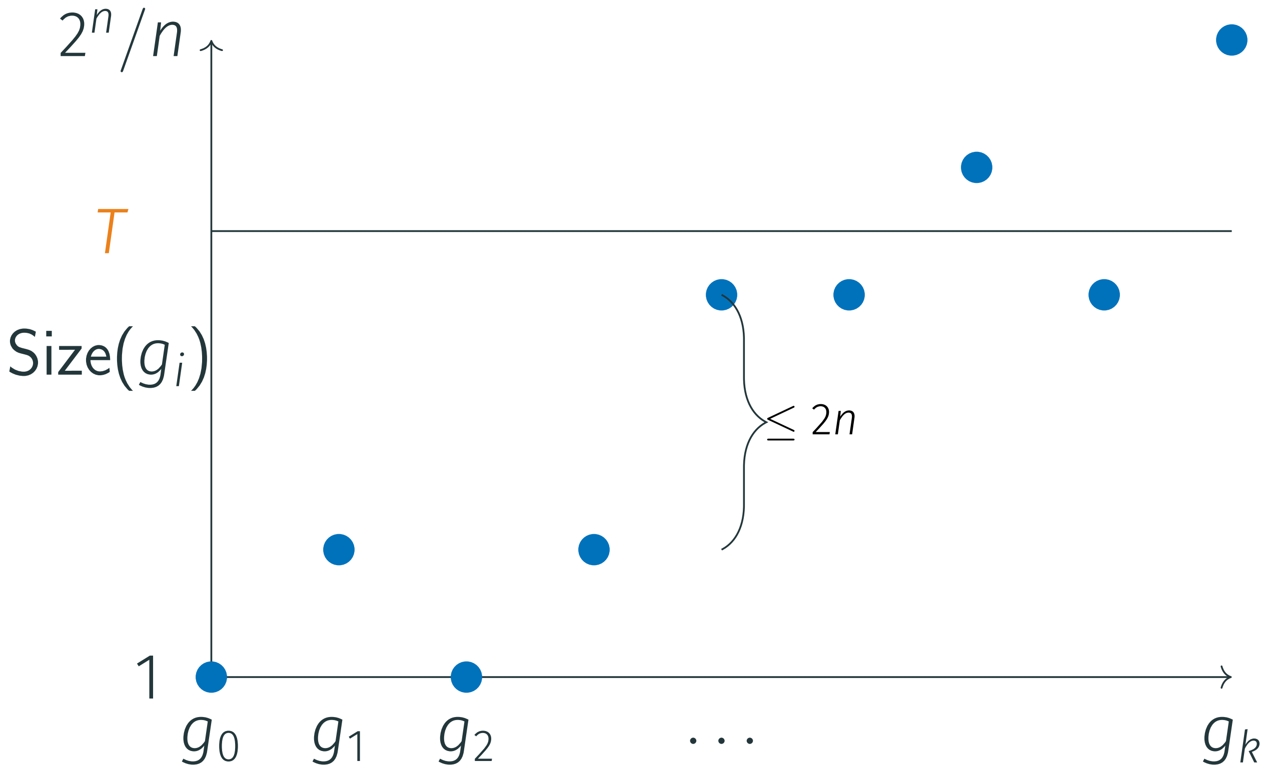
Hierarchy Theorem



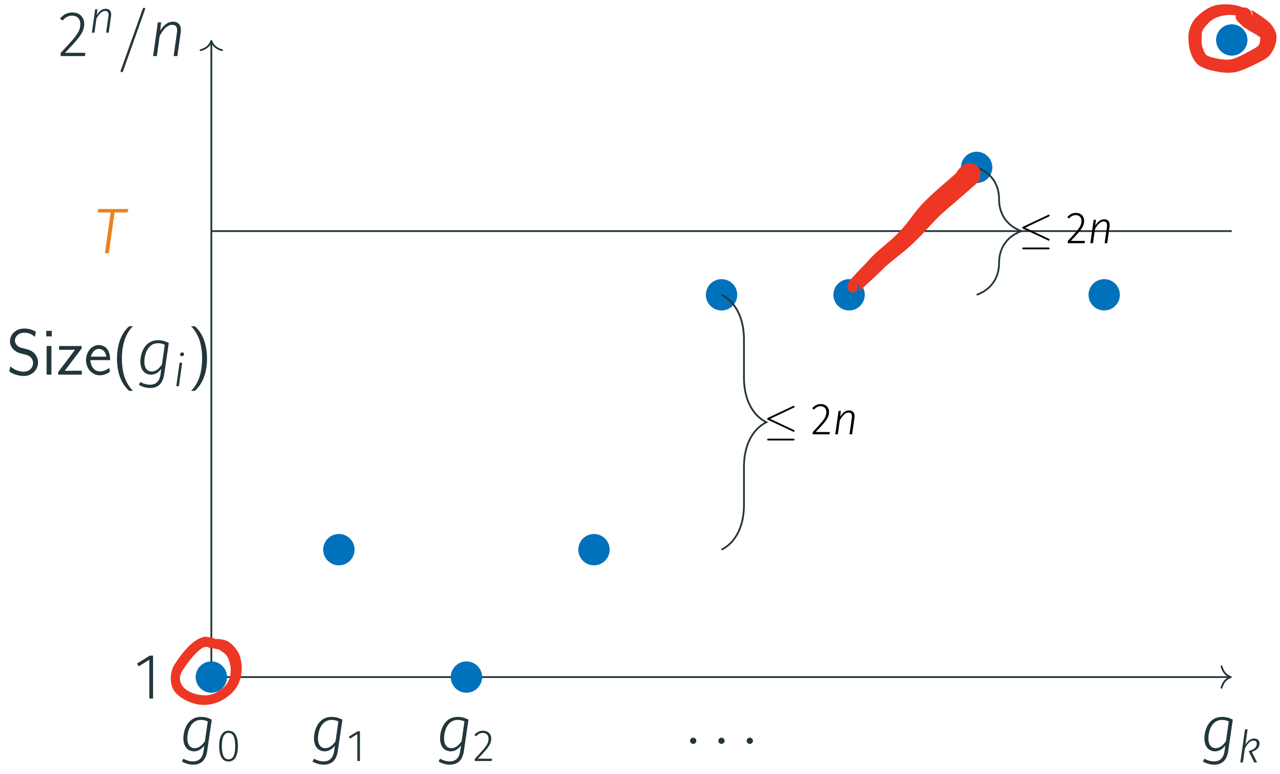
Hierarchy Theorem



Hierarchy Theorem



Hierarchy Theorem



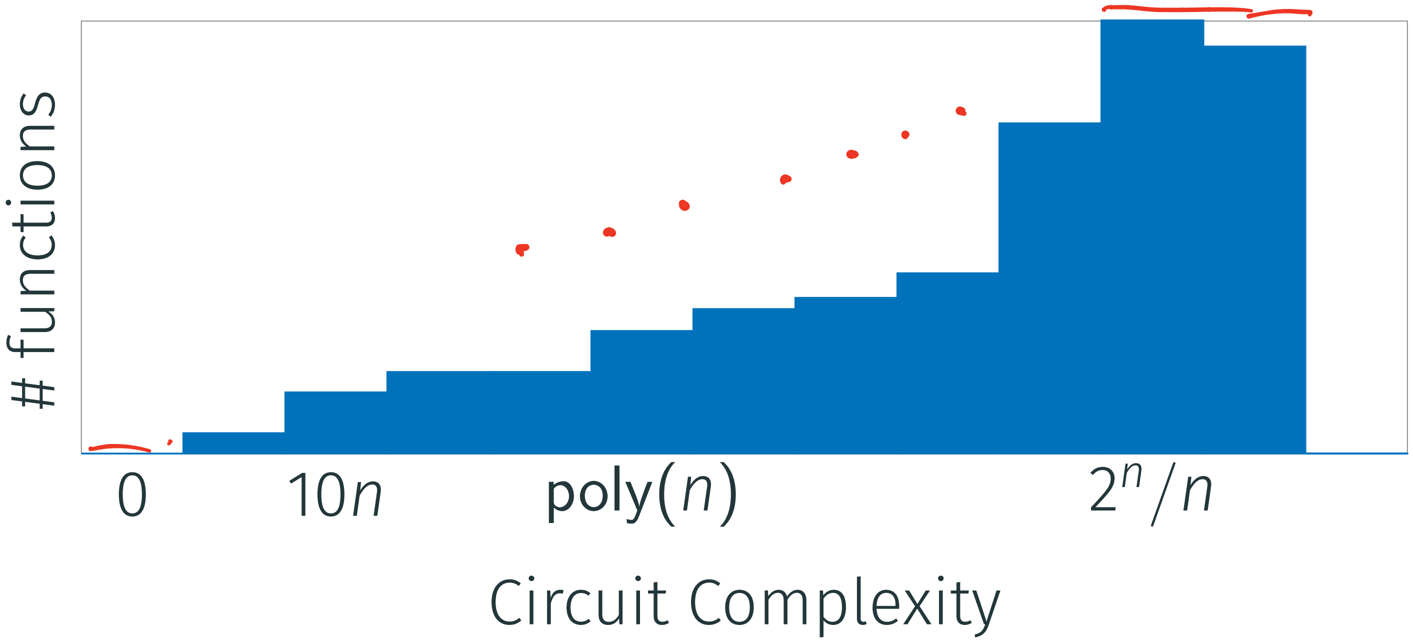
Hierarchy Theorem

Theorem

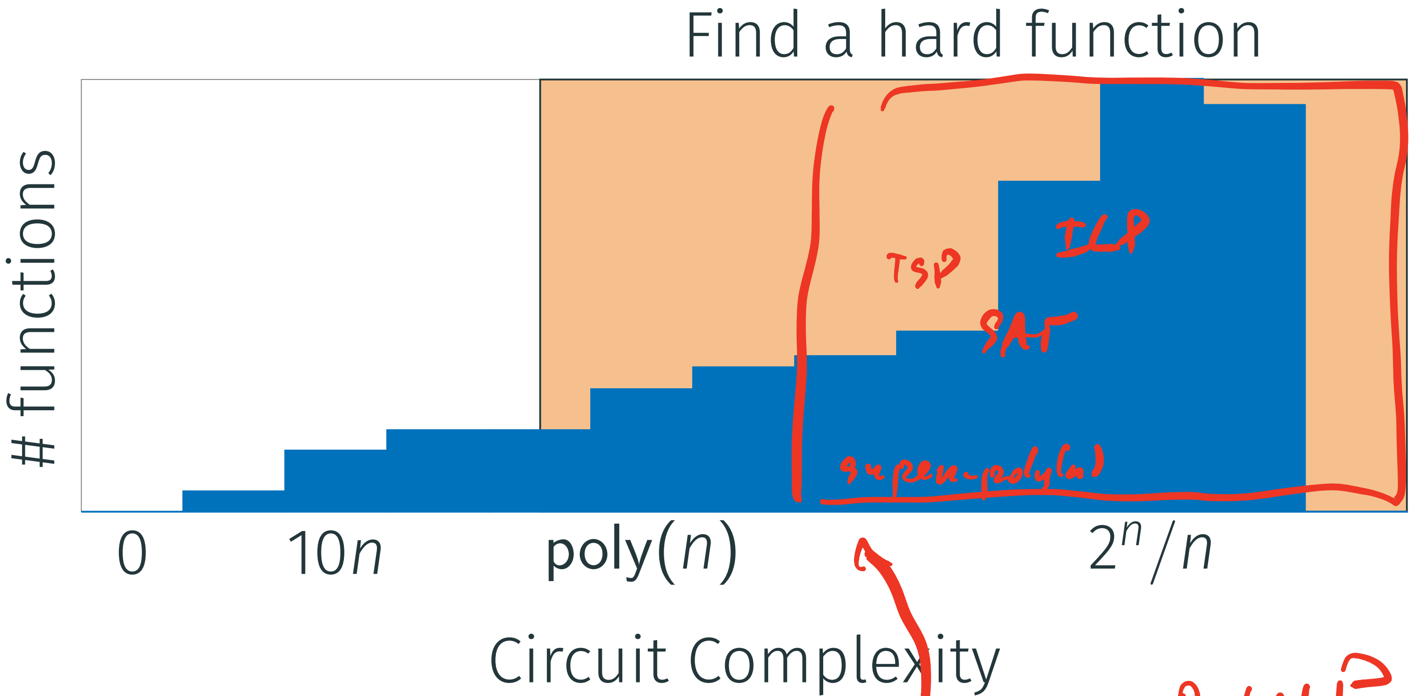
For any $T \leq 2^n/n$, there is a function $f\colon \{0,1\}^n \to \{0,1\}$ s.t.

$$\mathrm{Size}(f) = T \pm n\,.$$

Goal



Goal



Example, find $f \in NP$ $\Rightarrow$ $P \neq NP$

Circuit Complexity

- Goal: Find a hard function

Circuit Complexity

- Goal: Find a hard function
- Lower bounds: what functions are hard

Circuit Complexity

- Goal: Find a hard function
- Lower bounds: what functions are hard
- Upper bounds: what functions are easy

Circuit Upper Bound. Proof

Upper Bound [Lup1958]

Any function can be computed by a circuit of size

$$\frac{2^n}{n}$$

$$\leq 10 \cdot 2^n$$

simpler bound

Circuit Upper Bound. Proof

Upper Bound [Lup1958]

Any function can be computed by a circuit of size

$$\leq 10 \cdot 2^n$$

function of n-1 inputs

$$f(x_1, \ldots, x_n) = \begin{cases} f(1, x_2, \ldots, x_n), & \text{if } x_1 = 1 \\ f(0, x_2, \ldots, x_n), & \text{if } x_1 = 0 \end{cases}$$

function of n-1 inputs

Upper Bound [Lup1958]

Any function can be computed by a circuit of size

$$\leq 10 \cdot 2^n$$

$$f(x_1, \dots, x_n) = \boxed{\begin{cases} \underline{f(1, x_2, \dots, x_n)}, & \\ \underline{f(0, x_2, \dots, x_n)}, & \end{cases}} \begin{matrix} \text{if } x_1 = 1 \\ \text{if } x_1 = 0 \end{matrix}$$

$$= \cancel{(\overset{0}{x_1} \wedge f(1, x_2, \dots, x_n))} \vee (\cancel{\overset{1}{\overline{x}_1}} \wedge \underline{f(0, x_2, \dots, x_n)})$$

$x_1 = 0$

Circuit Upper Bound. Proof

Upper Bound [Lup1958]

Any function can be computed by a circuit of size

$$\boxed{\leq 10 \cdot 2^n}$$

$$\underline{f(x_1, \ldots, x_n)} = \begin{cases} f(1, x_2, \ldots, x_n), & \text{if } x_1 = 1 \\ f(0, x_2, \ldots, x_n), & \text{if } x_1 = 0 \end{cases}$$

$$= (x_1 \wedge f(1, x_2, \ldots, x_n)) \vee (\bar{x}_1 \wedge f(0, x_2, \ldots, x_n))$$

$$= (x_1 \boxed{\wedge} g_1(\underbrace{x_2, \ldots, x_n}_{n-1 \text{ inputs}})) \boxed{\vee} (\bar{x}_1 \boxed{\wedge} g_0(\underbrace{x_2, \ldots, x_n}_{n-1 \text{ inputs}}))$$

Any function with n inputs can be computed (i) using circuits for 2 functions with $n-1$ inputs (ii) and 4 additional gates

By induction, every of $n-1$ vars can be computed by a circuit of size $10 \cdot 2^{n-1}$

Circuit Upper Bound. Proof

Upper Bound [Lup1958]

Any function can be computed by a circuit of size

$$\leq \boxed{10 \cdot 2^n}$$

$$f(x_1, \dots, x_n) = \begin{cases} f(1, x_2, \dots, x_n), & \text{if } x_1 = 1 \\ f(0, x_2, \dots, x_n), & \text{if } x_1 = 0 \end{cases}$$

$$= (x_1 \wedge f(1, x_2, \dots, x_n)) \vee (\bar{x}_1 \wedge f(0, x_2, \dots, x_n))$$

$$= (x_1 \wedge g_1(x_2, \dots, x_n)) \vee (\bar{x}_1 \wedge g_0(x_2, \dots, x_n))$$

$$\mathrm{Size}(n) \leq 4 + 2\,\mathrm{Size}(n-1) = O(2^n)$$

$$\mathrm{size}(n) \le 4 + 2\,\mathrm{size}(n-1)$$

$$\mathrm{size}(1) = 1$$

$$\mathrm{size}(2) \le 6$$

$$\mathrm{size}(3) \le 16$$

$$\mathrm{size}(4) \le 36$$

$$\vdots$$

$$\underline{\mathrm{size}(n) \le 2.5 \cdot 2^{n} - 4}$$

$$\mathrm{size}(n-1) = 2.5 \cdot 2^{n-1} - 4$$

$$\mathrm{size}(n) = 4 + 2\,\mathrm{size}(n-1) =$$

$$= 4 + 2(2.5 \cdot 2^{n-1} - 4)$$

$$= \underline{2.5 \cdot 2^{n} - 4}$$

Circuit Lower Bound. Proof

Lower Bound [Sha1949]

Almost all functions of n variables have circuit size

$$\geq \boxed{2^n/(10n)}$$

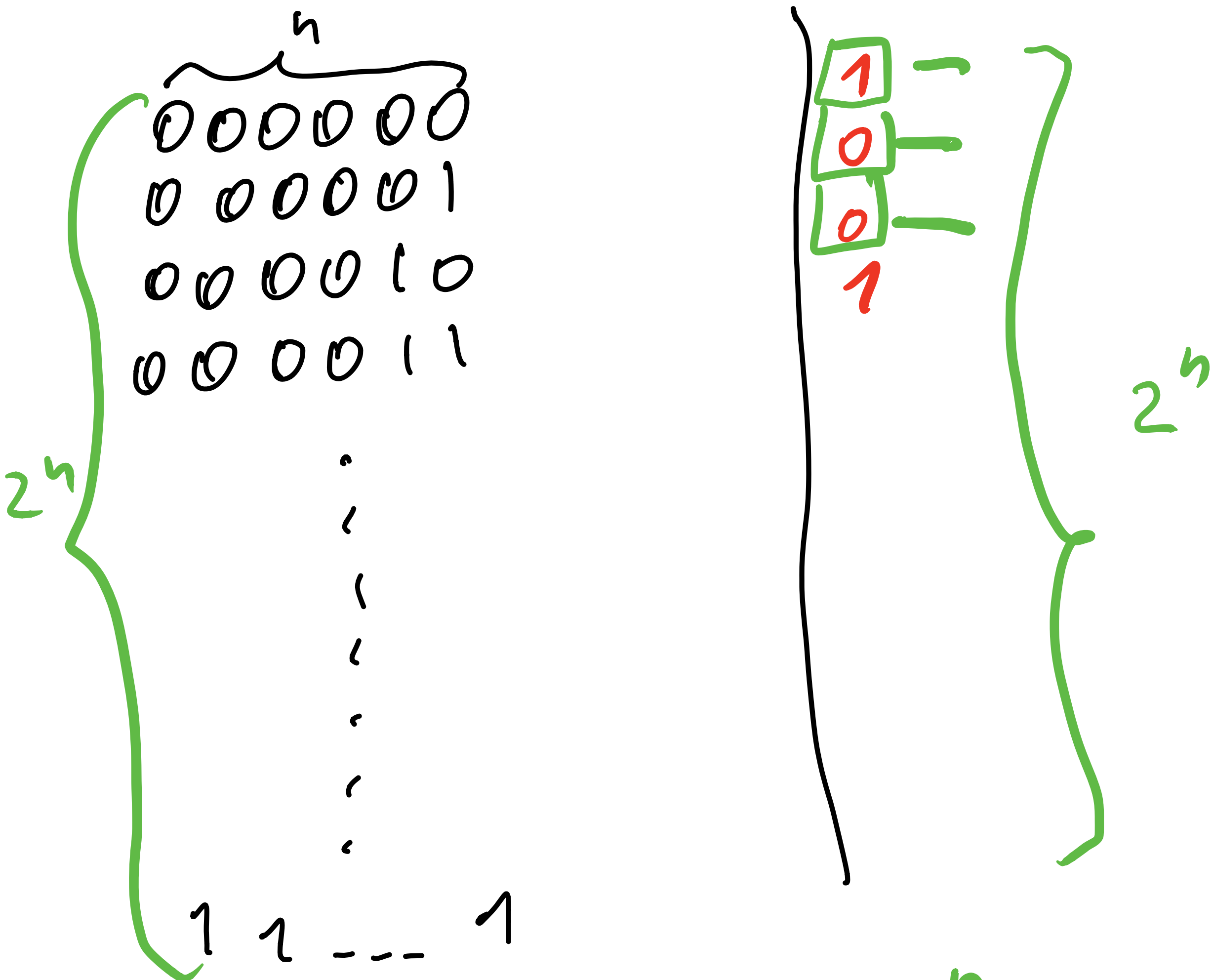
Lower Bound [Sha1949]

Almost all functions of n variables have circuit size

$$\geq 2^n/(10n)$$

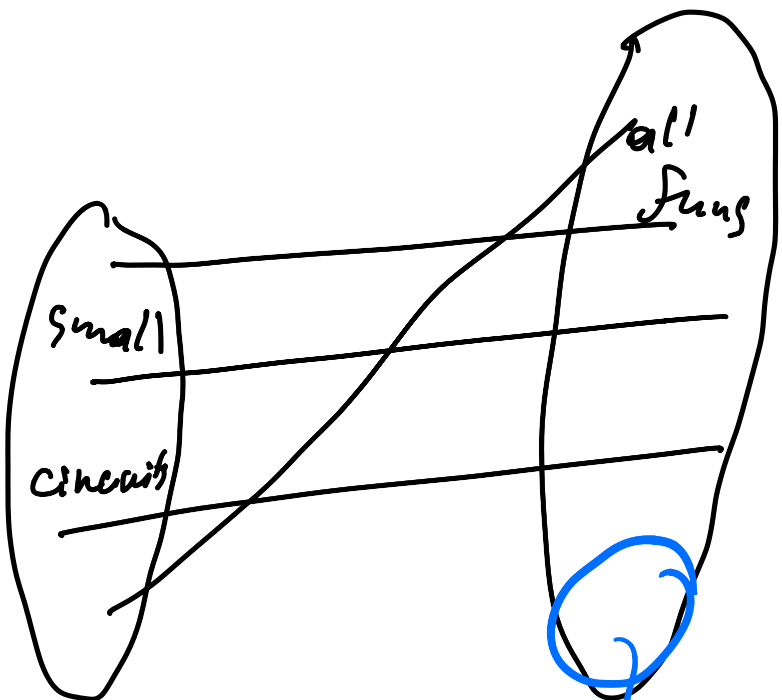
Let's count # of $f : \{0,1\}^n \to \{0,1\}$

Truth table ≡ table of its values



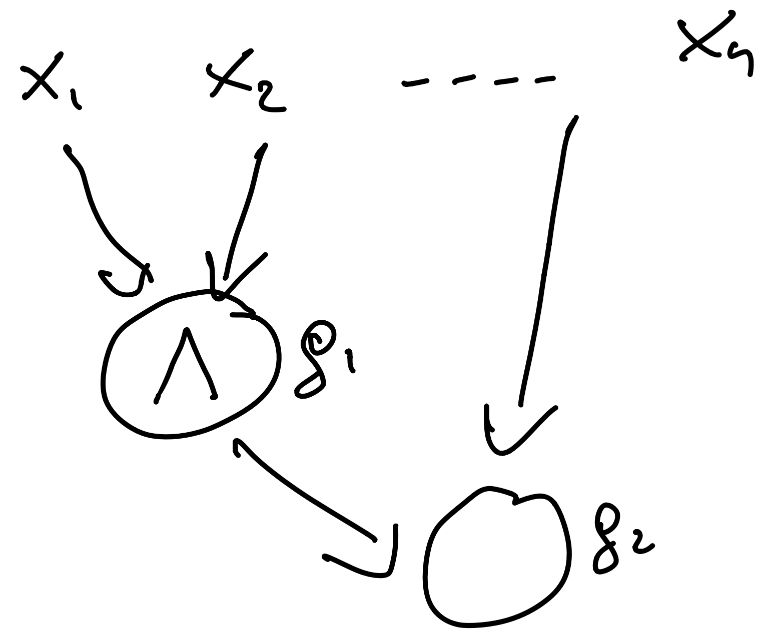
# of functions is 2^{2^n}

The number of circuits of size $\leq \boxed{\frac{2^n}{10n}}$ is $\ll 2^{2^n}$



cannot be computed
require large circuits

We want to show # of circuits of size $< \boxed{\frac{2^n}{10n}} = S$ is $\ll 2^{2^n}$



$$\left(3 \text{ functions} \cdot S \cdot S\right)^S$$

inputs of this gate

# of circuits $< (3S^2)^S <$

$= 2^{3S \log S}$

# of cuts of size $s = \boxed{\frac{2^n}{10n}}$

is $< 2^{3s\log s} < 2^{\frac{3}{10}\cdot 2^n} \ll$

2^{2^n} — # of functions

$\square$