# On lower bounds for circuit complexity and algorithms for satisfiability

April 2021

# Circuit lower bounds

We are interested in classifying the computational power of circuits. In particular we want to find for different types of circuits what are they limitations. Example:

$$\forall C \in \mathrm{AC}^0 \;\; |C| = O(n^k) \implies C \text{ cannot compute } \texttt{PARITY}$$

# Previous results

The circuit lower bounds area was most active during the 80's

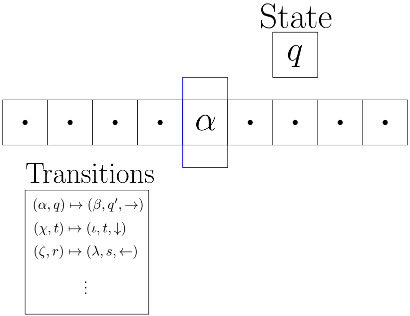
- $\mathsf{NEXP}^{\mathsf{NP}}$ requires superpolynomial circuits [Kannan '82]
- PARITY is not in $\mathsf{AC}^0$ [Ajtai '83, Håstad '86]
- PARITY with mod 3 gates is not in $\mathsf{ACC}^0$ [Razborov '87, Smolensky '87]

# Turing machine

A Turing machine is a tuple composed of

- An alphabet $\Gamma$
- A set of states $Q$
- A function $\delta : \Gamma \times Q \mapsto \Gamma \times Q \times H$ where $H = \{\text{left}, \text{stay}, \text{right}\}$

We can be interested in the number of steps they take (time) or the amount of tape cells they use (space).

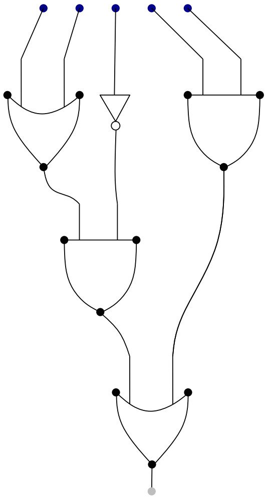


# Circuits

Circuits are DAGs where

- $V = \{v_1, \dots, v_k\}$ with $v_i \in \{I, O, \wedge, \vee, \neg\}$.
- $E \subseteq V \times V$.

A circuit has exactly $n$ input vertices and 1 output vertex. We can be interested in the number of vertices (size) or the longest path from input to output (depth).



# Complexity class

A complexity class C is a collection of sets $\{A_1, A_2, A_3 \ldots\}$ with $A_i \subseteq \mathbb{N}$, such that computing $\chi(x, A_i)$ (the charachteristic function of $A_i$) takes a "similar" amount of resources between all $i$. We usually call the $A_i$'s "languages".

# Complexity class

A complexity class C is a collection of sets $\{A_1, A_2, A_3 \ldots\}$ with $A_i \subseteq \mathbb{N}$, such that computing $\chi(x, A_i)$ (the charachteristic function of $A_i$) takes a "similar" amount of resources between all $i$. We usually call the $A_i$'s "languages".

Some useful classes

- $\mathsf{NEXP} = \bigcup_{c>0} \mathsf{NTIME}(2^{n^c})$.
- $\mathsf{P}/poly = \bigcup_{c>0} \mathsf{SIZE}(n^c)$.
- $\mathsf{PSPACE} = \bigcup_{c>0} \mathsf{SPACE}(n^c)$.
- MA.

# Verifier

A verifier $V$ for a language $L$ is a polynomial time Turing machine such that on input $x$

- If $x \in L$ then there exists $y \in \{0,1\}^{t(n)}$ such that $V(x,y) = 1$ where $t(n)$ depends on $L$
- If $x \notin L$ then for every $y \in \{0,1\}^*$ $V(x,y) = 0$

# Universal "Small" witness circuits

A witness is the string $y$ with which the a verifier V certifies the membership of $x$ in $L$. A circuit $C$ is a witness circuit if the string $z$ defined as

$$\forall\, i \in \{1, \ldots, t(n)\} \quad z_i = C(x, i)$$

implies $V(x, z) = 1$. For us a circuit will be "small" if it has polynomial size.

$L$ has universal "small" witness $\iff$ for all correct $V$ we have such $C$

# Williams' method

Williams' method yields the conditional lower bound NEXP $\not\subseteq$ P/*poly*. The method has two parts:

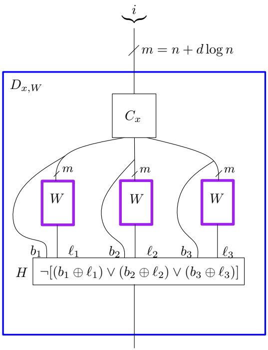
- (A) If NEXP $\subseteq$ P/*poly* then there exists universal "small" witness circuits
- (B) If there exists a better-than-trivial algorithm for `CIRCUIT SAT` then there cannot exist universal "small" witness circuits

# Williams' method II

Part (A) yields the witness circuits W of the appropriate size. Part (B) says that unsatisfiability of $D_{x,W}$ can be decided "fast" using W.

$$D_{x,W} \in \texttt{UNSAT} \iff x \in L$$

For appropriate $L$, we get a contradiction.



# Outline

We will show that if NEXP $\subseteq$ P/*poly* and there exists $L \in$ NEXP without universal "small" witness circuits we are lead to the following inclusion:

$$\text{EXP} \subseteq \text{io-SIZE}(n^q)$$

# Outline

We will show that if NEXP $\subseteq$ P/*poly* and there exists $L \in$ NEXP without universal "small" witness circuits we are lead to the following inclusion:

$$\underset{\text{PSPACE}}{\not{\text{EXP}}} \subseteq \text{io-SIZE}(n^q)$$

PSPACE $\subseteq$ io-SIZE$(n^q)$ is a contradiction (proof by diagonalization)

# Outline

The proof of the inclusion

$$\text{PSPACE} \subseteq \text{io-SIZE}(n^q)$$

is divided in three inclusions:

- PSPACE $\subseteq$ MA
- MA $\subseteq$ io-NTIME$(2^n)/n$
- io-NTIME$(2^n)/n \subseteq$ io-SIZE$(n^q)$

# A note on pseudorandomness

Let $x \in L$ and suppose that $L$ does not have universal "small" witness circuits.
Then for some $V$ and $y \in \{0,1\}^*$, such that $V(x,y) = 1$ we have that for any circuit $C$ with $|C| \leq n^k$

$$\exists z \text{ such that } C(z) \neq y_z$$

# A note on pseudorandomness

Let $x \in L$ and suppose that $L$ does not have universal "small" witness circuits.
Then for some $V$ and $y \in \{0,1\}^*$, such that $V(x,y) = 1$ we have that for any circuit $C$ with $|C| \leq n^k$

$$\exists z \text{ such that } C(z) \neq y_z$$

$y$ is the truth table of a "hard" function $\implies$ Can construct a pseudorandom generator.

# The proof

**Assumption**: NEXP $\subseteq$ P/*poly* and there exists $L \in$ NEXP that does not have universal polynomial size witness circuits.

We must prove:

- PSPACE $\subseteq$ MA
- MA $\subseteq$ io-NTIME$(2^n)/n$
- io-NTIME$(2^n)/n \subseteq$ io-SIZE$(n^q)$

Putting everything together:

$$\text{PSPACE} \subseteq \text{MA} \subseteq \text{io-NTIME}(2^n)/n \subseteq \text{io-SIZE}(n^q)$$

for constant $q$.

# The proof

**Assumption**: NEXP $\subseteq$ P/*poly* and there exists $L \in$ NEXP that does not have universal polynomial size witness circuits.
We must prove:

- PSPACE $\subseteq$ MA easy simulation
- MA $\subseteq$ io-NTIME$(2^n)/n$ using witness as hard function
- io-NTIME$(2^n)/n \subseteq$ io-SIZE$(n^q)$ careful simulation

Putting everything together:

$$\text{PSPACE} \subseteq \text{MA} \subseteq \text{io-NTIME}(2^n)/n \subseteq \text{io-SIZE}(n^q)$$

for constant $q$.

# The proof

Since the inclusion PSPACE $\subseteq$ io-SIZE($n^q$) is false, we get that

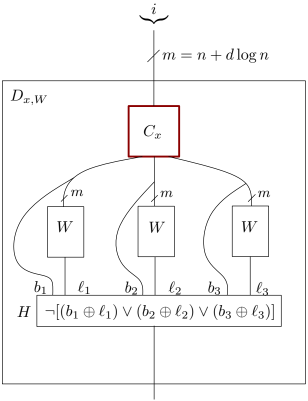
$$\text{NEXP} \subseteq \text{P}/poly \implies \text{NEXP has universal "small" witnesses}$$

# An unsatisfiable circuit

Fix $L \in \mathrm{NTIME}(2^n)$. Let $x$ with $|x| = n$ be an input. Suppose that $C_x$ encodes a Boolean formula $\Phi_x$ such that $\Phi_x \in \mathtt{SAT} \iff x \in L$, and $W$ is a witness circuit for some correct $V$. Then

$$D_{x,W} \in \mathtt{UNSAT} \iff x \in L$$

The input is the index of a clause of $\Phi_x$ and $d$ is a constant independent of $L$ and $x$.



# No universal "small" witness circuits

Pick $L \in \mathrm{NTIME}(2^n) \setminus \mathrm{NTIME}(2^{n-\omega(\log n)})$ (which exists by the non-deterministic time hierarchy). Build $D_{x,W}$. Suppose that CIRCUIT SAT can be solved in time

$$O\left(\frac{2^n \cdot (n^{k^*})^c}{f(n)}\right) = O(2^{n+c\cdot k^* \log n - \omega(\log n)})$$

where $f(n)$ is superpolynomial. Then, we could decide $L$ in time $O(2^{n-\omega(\log n)})$, a contradiction.

# A first reduction

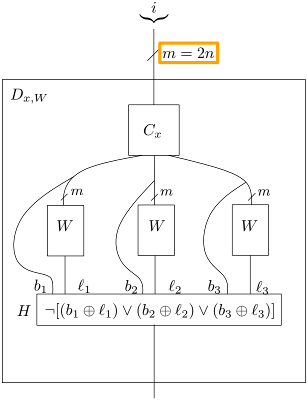
The Cook-Levin theorem offers a construction such that for a fixed language $L$, given an input $x$ there exists a Boolean formula $\Psi_x$ such that

$$\Psi_x \in \mathtt{SAT} \iff x \in L$$

and $|\Psi_x| = O(n^2)$. Moreover the $i$-th clause of $\Psi_x$ can be computed in time $O(\log^{O(1)} n)$.

# A first reduction II

Thus, we get $C_x$ of size $O((\log^{O(1)} 2^n)^2) = O(n^k)$ (as needed) but with $2n$ inputs, which would make $D_{x,W}$ have $2n$ inputs.



# A first reduction III

If we apply the previous reasoning, we can decide the membership to $L$ in time

$$O\left(\frac{2^{2n} \cdot (n^{k^*})^c}{f(n)}\right) = O(2^{2n+c \cdot k^* \log n - \omega(\log n)})$$

Not necessarily $O(2^{n-\omega(\log n)})$

# A more efficient reduction

We can use quite old work from Stearns & Hunt and from Robson to construct a formula $\Phi_x$ with

$$\Phi_x \in \mathtt{SAT} \iff x \in L$$

and $|\Phi_x| = O(n \log^{O(1)} n)$. The $i$-th clause of $\Phi_x$ is also computable in time $O(log^{O(1)} n)$.

# Summary

Fix $L \in \mathrm{NTIME}(2^n) \setminus \mathrm{NTIME}(2^{n-\omega(\log n)})$. Assuming that $\mathrm{NEXP} \subseteq \mathrm{P}/poly$ we get that $L$ has universal "small" witness circuits. Construct $D_{x,W}$ and execute the better-than-trivial algorithm for CIRCUIT SAT with input $D_{x,W}$. Thus, decide $L$ in time $O(2^{n-\omega(\log n)})$, a contradiction.

# Relating two branches

In principle, proving circuit lower bounds and designing algorithms need not be related

- The former concerns showing that **for all** circuits some function is not computable
- The latter concerns showing that **there exists** a circuit that computes some function

# Future work

Interesting research paths after this work:

- Produce and/or publish a **complete** proof of the construction of $C_x$.
- Can we use other NP-complete problems to construct $C_x$? (Maybe more efficient).
- Consider sorting networks to construct the efficient reduction for $C_x$.
- Thorough study of Williams' method against complexity barriers

The presentation has finished