

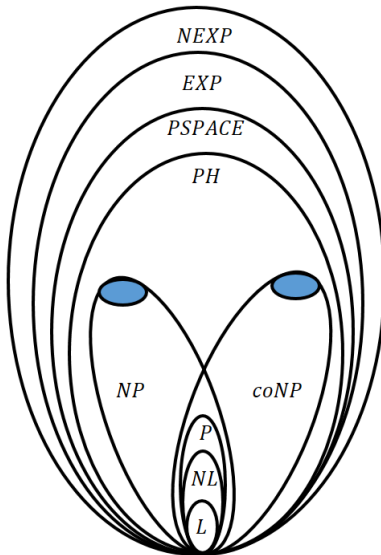
CSL853: Complexity Theory

Ragesh Jaiswal, CSE, IIT Delhi

Complexity Diagram

Complexity Diagram

Our current view of complexity classes



Boolean Circuits

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Definition (Boolean Circuits)

For every $n \in \mathbb{N}$, an n -input, single-output Boolean circuit is a directed graph with n *sources* (vertices with no incoming edges) and one *sink* (vertex with no outgoing edges). All non-source vertices are called *gates* and are labeled with one of $\wedge$, $\vee$, or $\neg$. The vertices labeled with $\wedge$ and $\vee$ have *fan-in* equal to 2 and the vertices labeled with $\neg$ have fan-in 1. The *size* of C , denoted by $|C|$, is the number of vertices in it.

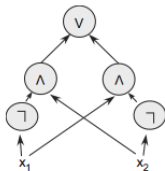


Figure : Boolean circuit that computes XOR of two bits

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Definition (Boolean Circuits)

For every $n \in \mathbb{N}$, an n -input, single-output Boolean circuit is a directed graph with n *sources* (vertices with no incoming edges) and one *sink* (vertex with no outgoing edges). All non-source vertices are called *gates* and are labeled with one of $\wedge$, $\vee$, or $\neg$. The vertices labeled with $\wedge$ and $\vee$ have *fan-in* equal to 2 and the vertices labeled with $\neg$ have fan-in 1. The *size* of C , denoted by $|C|$, is the number of vertices in it.

Definition (Circuit family)

Let $T : \mathbb{N} \rightarrow \mathbb{N}$ be a function. A $T(n)$ size circuit family is a sequence $\{C_n\}_{n \in \mathbb{N}}$ of boolean Circuits, where C_n has n inputs and a single output, and its size $|C_n| \leq T(n)$ for every n .

Definition (Language recognition by circuit family)

We say that a language L is in $\text{SIZE}(T(n))$ if there exists a $T(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every $x \in \{0, 1\}^n$, $x \in L \Leftrightarrow C_n(x) = 1$.

Boolean Circuits

Boolean Circuits and P_{poly}

- Uniform Vs Non-uniform
 - Turing machine is *uniform* model of computation.
 - The notion of decidability of a language $L \subseteq \{0, 1\}^*$ is with respect to a single Turing Machine. That is, a single machine should work for all input sizes.
 - Boolean circuit family represent a *non-uniform* model of computation.
 - Within this model, we are allowed to have a different circuit for every input size.

Boolean Circuits

Boolean Circuits and P_{poly}

Definition (Circuit family)

Let $T : \mathbb{N} \rightarrow \mathbb{N}$ be a function. A $T(n)$ size circuit family is a sequence $\{C_n\}_{n \in \mathbb{N}}$ of boolean Circuits, where C_n has n inputs and a single output, and its size $|C_n| \leq T(n)$ for every n .

Definition (Language recognition by circuit family)

We say that a language L is in $\text{SIZE}(T(n))$ if there exists a $T(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every $x \in \{0, 1\}^n$, $x \in L \Leftrightarrow C_n(x) = 1$.

- Consider the following languages:
 - $L_1 = \{1^n : n \in \mathbb{Z}\}$
 - $L_2 = \{\langle m, n, m + n \rangle : m, n \in \mathbb{Z}\}$
- Show that $L_1 \in \text{SIZE}(O(n))$.
- Show that $L_2 \in \text{SIZE}(O(n))$

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Definition (Circuit family)

Let $T : \mathbb{N} \rightarrow \mathbb{N}$ be a function. A $T(n)$ size circuit family is a sequence $\{C_n\}_{n \in \mathbb{N}}$ of boolean Circuits, where C_n has n inputs and a single output, and its size $|C_n| \leq T(n)$ for every n .

Definition (Language recognition by circuit family)

We say that a language L is in $\text{SIZE}(T(n))$ if there exists a $T(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every $x \in \{0, 1\}^n$,
 $x \in L \Leftrightarrow C_n(x) = 1$.

- Consider the following languages:
 - $L_1 = \{1^n : n \in \mathbb{Z}\}$
 - $L_2 = \{\langle m, n, m + n \rangle : m, n \in \mathbb{Z}\}$
- Show that $L_1 \in \text{SIZE}(O(n))$.
- Show that $L_2 \in \text{SIZE}(O(n))$
- Show that any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ can be computed by a Boolean circuit of size $n \cdot 2^n$.

Boolean Circuits

Boolean Circuits and $P_{/\text{poly}}$

Definition (The class $P_{/\text{poly}}$)

$P_{/\text{poly}}$ is the class of languages that are decidable by polynomial-size circuit families. That is, $P_{/\text{poly}} = \cup_c \text{SIZE}(n^c)$.

- How are P and $P_{/\text{poly}}$ related?

Boolean Circuits

Boolean Circuits and $P_{/\text{poly}}$

Definition (The class $P_{/\text{poly}}$)

$P_{/\text{poly}}$ is the class of languages that are decidable by polynomial-size circuit families. That is, $P_{/\text{poly}} = \cup_c \text{SIZE}(n^c)$.

Theorem

$P \subseteq P_{/\text{poly}}$.

Proof.

Proof follows from Cook-Levin theorem. □

Boolean Circuits

Boolean Circuits and $P_{/\text{poly}}$

Definition (The class $P_{/\text{poly}}$)

$P_{/\text{poly}}$ is the class of languages that are decidable by polynomial-size circuit families. That is, $P_{/\text{poly}} = \bigcup_c \text{SIZE}(n^c)$.

Theorem

$P \subseteq P_{/\text{poly}}$.

- Is $P \subsetneq P_{/\text{poly}}$?

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem

$$P \subsetneq P_{/poly}.$$

Proof.

- Claim 1: Let $L \subseteq \{0,1\}^*$ be any unary language (i.e., $L \subseteq \{1^n : n \in \mathbb{N}\}$). Then $L \in P_{/poly}$.
- Claim 2: There exists a unary language that is undecidable.
 - Consider the following language:

$$\text{UHALT} = \{1^n : n\text{'s binary expansion encodes a pair } \langle M, x \rangle \text{ such that } M \text{ halts on input } x\}$$

- Note that UHALT is undecidable.



Boolean Circuits

Boolean Circuits and P_{poly}

- The set of languages decided by *arbitrary* circuit families may contain arbitrary hard problems as we saw in the previous slide.
- What about circuit families that can be *efficiently generated*?

Definition (P-uniform circuit families)

A circuit family $\{C_n\}$ is P-uniform if there is a polynomial-time TM that on input 1^n outputs the description of the circuit C_n .

Boolean Circuits

Boolean Circuits and $P_{/poly}$

- The set of languages decided by *arbitrary* circuit families may contain arbitrary hard problems as we saw in the previous slide.
- What about circuit families that can be *efficiently generated*?

Definition (P-uniform circuit families)

A circuit family $\{C_n\}$ is P-uniform if there is a polynomial-time TM that on input 1^n outputs the description of the circuit C_n .

Theorem

A language L is computable by a P-uniform circuit family iff $L \in P$.

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Definition (P-uniform circuit families)

A circuit family $\{C_n\}$ is P-uniform if there is a polynomial-time TM that on input 1^n outputs the description of the circuit C_n .

Theorem

A language L is computable by a P-uniform circuit family iff $L \in P$.

- We may similarly define *logspace*-uniform circuit families.

Definition (logspace-uniform circuit families)

A circuit family $\{C_n\}$ is logspace uniform if there is an implicitly logspace computable function mapping 1^n to the description of the circuit C_n .

Theorem

A language has logspace-uniform circuits of polynomial size iff it is in P.

Boolean Circuits

Boolean Circuits and $P_{/\text{poly}}$

Definition (Turing Machines with Advice)

Let $T, a : \mathbb{N} \rightarrow \mathbb{N}$ be functions. The class of languages decidable by time- $T(n)$ TM's with $a(n)$ advice, denoted $\text{DTIME}(T(n))_{/a(n)}$, contains every L such that there exists a sequence $\{\alpha_n\}_{n \in \mathbb{N}}$ of strings with $\alpha_n \in \{0, 1\}^{a(n)}$ and a TM M satisfying $M(x, \alpha_n) = 1 \Leftrightarrow x \in L$ for every $x \in \{0, 1\}^n$, where on input (x, α_n) the machine M runs for at most $O(T(n))$ steps.

Theorem

$$P_{/\text{poly}} = \bigcup_{c,d} \text{DTIME}(n^c)_{/n^d}.$$

Boolean Circuits

Boolean Circuits and $P_{/\text{poly}}$

Definition (Turing Machines with Advice)

Let $T, a : \mathbb{N} \rightarrow \mathbb{N}$ be functions. The class of languages decidable by time- $T(n)$ TM's with $a(n)$ advice, denoted $\text{DTIME}(T(n))_{/a(n)}$, contains every L such that there exists a sequence $\{\alpha_n\}_{n \in \mathbb{N}}$ of strings with $\alpha_n \in \{0, 1\}^{a(n)}$ and a TM M satisfying $M(x, \alpha_n) = 1 \Leftrightarrow x \in L$ for every $x \in \{0, 1\}^n$, where on input (x, α_n) the machine M runs for at most $O(T(n))$ steps.

Theorem

$$P_{/\text{poly}} = \bigcup_{c,d} \text{DTIME}(n^c)_{/n^d}.$$

- What is the relationship of $P_{/\text{poly}}$ with NP?

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{/poly}$, then $PH = \Sigma_2^P$.

Proof.

- Claim 1: It is sufficient to show that $\Pi_2^P \subseteq \Sigma_2^P$.

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{/poly}$, then $PH = \Sigma_2^P$.

Proof.

- Claim 1: It is sufficient to show that $\Pi_2^P \subseteq \Sigma_2^P$.
- We will show that $\Pi_2SAT = \overline{\Sigma_2SAT}$ consisting of true formulas of the form $\forall u \in \{0,1\}^n \exists v \in \{0,1\}^n \psi(u,v) = 1$ belongs to Σ_2^P .

Boolean Circuits

Boolean Circuits and P_{poly}

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{\text{poly}}$, then $PH = \Sigma_2^P$.

Proof.

- Claim 1: It is sufficient to show that $\Pi_2^P \subseteq \Sigma_2^P$.
- We will show that $\Pi_2\text{SAT} = \overline{\Sigma_2\text{SAT}}$ consisting of true formulas of the form $\forall u \in \{0, 1\}^n \exists v \in \{0, 1\}^n \psi(u, v) = 1$ belongs to Σ_2^P .
- Since $NP \subseteq P_{\text{poly}}$, there is a polynomial p and a $p(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every boolean formula ψ and $u \in \{0, 1\}^n$, $C_n(\psi, u) = 1$ iff there exists a $v \in \{0, 1\}^n$ s.t. $\psi(u, v) = 1$.

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{/poly}$, then $PH = \Sigma_2^P$.

Proof.

- Claim 1: It is sufficient to show that $\Pi_2^P \subseteq \Sigma_2^P$.
- We will show that $\Pi_2SAT = \overline{\Sigma_2SAT}$ consisting of true formulas of the form $\forall u \in \{0, 1\}^n \exists v \in \{0, 1\}^n \psi(u, v) = 1$ belongs to Σ_2^P .
- Since $NP \subseteq P_{/poly}$, there is a polynomial p and a $p(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every boolean formula ψ and $u \in \{0, 1\}^n$, $C_n(\psi, u) = 1$ iff there exists a $v \in \{0, 1\}^n$ s.t. $\psi(u, v) = 1$.
- Claim 2: There is a polynomial q and a $q(n)$ -size circuit family $\{C'_n\}_{n \in \mathbb{N}}$ s.t. for every boolean formula ψ and $u \in \{0, 1\}^n$, it outputs $v \in \{0, 1\}^n$ such that $\psi(u, v) = 1$ given that such a v exists.

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{/poly}$, then $PH = \Sigma_2^P$.

Proof.

- Claim 1: It is sufficient to show that $\Pi_2^P \subseteq \Sigma_2^P$.
- We will show that $\Pi_2SAT = \overline{\Sigma_2SAT}$ consisting of true formulas of the form $\forall u \in \{0,1\}^n \exists v \in \{0,1\}^n \psi(u, v) = 1$ belongs to Σ_2^P .
- Since $NP \subseteq P_{/poly}$, there is a polynomial p and a $p(n)$ -size circuit family $\{C_n\}_{n \in \mathbb{N}}$ such that for every boolean formula ψ and $u \in \{0,1\}^n$, $C_n(\psi, u) = 1$ iff there exists a $v \in \{0,1\}^n$ s.t. $\psi(u, v) = 1$.
- Claim 2: There is a polynomial q and a $q(n)$ -size circuit family $\{C'_n\}_{n \in \mathbb{N}}$ s.t. for every boolean formula ψ and $u \in \{0,1\}^n$, it outputs $v \in \{0,1\}^n$ such that $\psi(u, v) = 1$ given that such a v exists.
- Claim 3: $\forall u \in \{0,1\}^n \exists v \in \{0,1\}^n \psi(u, v) = 1 \Leftrightarrow$

$$\exists v \in \{0,1\}^{10q(n)^2} \forall u \in \{0,1\}^n \text{ s.t.}$$

$$w \text{ describes a circuit } C' \text{ and } \psi(u, C'(\psi, u)) = 1$$



Boolean Circuits

Boolean Circuits and P_{poly}

Theorem (Meyer's Theorem)

If $\text{EXP} \subseteq P_{\text{poly}}$, then $\text{EXP} = \Sigma_2^P$.

Boolean Circuits

Boolean Circuits and $P_{/poly}$

Theorem (Karp-Lipton Theorem, Karp, Lipton, 1980)

If $NP \subseteq P_{/poly}$, then $PH = \Sigma_2^P$.

- Show that if $NP \not\subseteq P_{/poly}$, then $P \neq NP$.
- So, to show that $P \neq NP$, we can alternatively show that there is an NP language that does not have a polynomial size circuit family.
- So, we should focus on *circuit lower bounds*.

Boolean Circuits

Circuit lower bounds

- Are there boolean functions that cannot be computed by small size circuits?

Theorem

For every $n > 1$, there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ that cannot be computed by a circuit C of size $\frac{2^n}{10n}$.

Boolean Circuits

Circuit lower bounds

Theorem

For every $n > 1$, there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ that cannot be computed by a circuit C of size $\frac{2^n}{10n}$.

- Are there boolean functions that cannot be computed by small size circuits? **Yes**.
- In fact, there are a huge number of functions that cannot be computed by any circuit of small size.

Theorem

For every $n > 1$, there exists a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ that cannot be computed by a circuit C of size $\frac{2^n}{10n}$.

- Are there boolean functions that cannot be computed by small size circuits? **Yes.**
- In fact, there are a huge number of functions that cannot be computed by any circuit of small size.
- And yet, we have not been able to find such an NP function until now!

End