אוניברסיטת
אריאל
בשומרון

Computational Complexity

Lecturer: Edward A. Hirsch

https://edwardahirsch.github.io/edwardahirsch

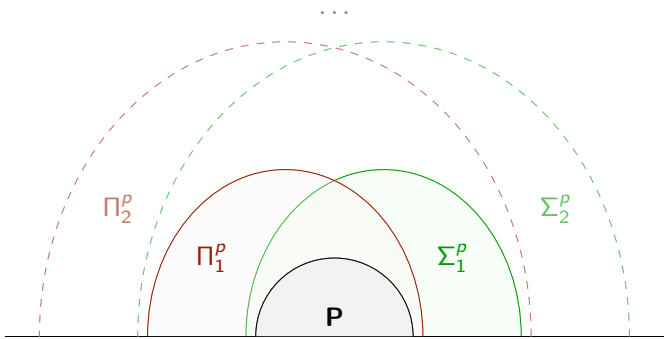
edward.a.hirsch@gmail.com

November 26, 2025

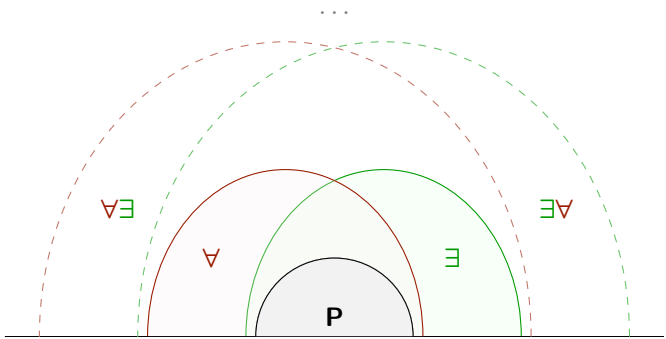
Lecture plan

- Polynomial Hierarchy:
  - Two definitions: equivalence.
  - Collapse.
  - Complete problems (and their absence).
  - Polynomial-size circuits and Karp–Lipton collapse (if time permits).

PH: Polynomial Hierarchy



PH: Polynomial Hierarchy



The oracle definition — and the one with quantifiers

A reminder from the previous lecture

Definition (classes of the polynomial hierarchy)

$$\Sigma_0^p = \Pi_0^p = \Delta_0^p = \mathbf{P}$$

$$\Sigma_1^p = \mathbf{NP}$$
$$\Pi_1^p = \mathbf{co\text{-}NP}$$
$$\Delta_1^p = \mathbf{P}$$

$$\Sigma_k^p = \mathbf{NP}^{\Pi_{k-1}^p} = \mathbf{NP}^{\Sigma_{k-1}^p}$$
$$\Pi_k^p = \mathbf{co\text{-}}\Sigma_k^p$$
$$\Delta_k^p = \mathbf{P}^{\Sigma_{k-1}^p}$$

Theorem

$\forall k \geqslant 1$
$L \in \Sigma_k^p \iff$ there is a p-bounded $T \in \Pi_{k-1}^p$ such that $\forall x$

$$x \in L \iff \exists w \ \langle x, w \rangle \in T,$$

The oracle definition — and the one with quantifiers

A reminder from the previous lecture

Definition (classes of the polynomial hierarchy)

$$
\begin{aligned}
\Sigma_0^p &= \Pi_0^p = \Delta_0^p = \mathbf{P} \\
\Sigma_1^p &= \mathbf{NP} \\
\Pi_1^p &= \mathbf{co\text{-}NP} \\
\Delta_1^p &= \mathbf{P} \\
\Sigma_k^p &= \mathbf{NP}^{\Pi_{k-1}^p} = \mathbf{NP}^{\Sigma_{k-1}^p} \\
\Pi_k^p &= \mathbf{co\text{-}}\Sigma_k^p \\
\Delta_k^p &= \mathbf{P}^{\Sigma_{k-1}^p}
\end{aligned}
$$

Theorem

$\forall k \geqslant 1$

$L \in \Sigma_k^p \iff$ there is a p-bounded $T \in \Pi_{k-1}^p$ such that $\forall x$

$$x \in L \iff \exists w \ \langle x, w \rangle \in T,$$

$L \in \Pi_k^p \iff$ there is a p-bounded $S \in \Sigma_{k-1}^p$ such that $\forall x$

$$x \in L \iff \forall w \ \langle x, w \rangle \in S$$

Corollary

$\forall k \geqslant 1$

$L \in \Sigma_k^p \iff$ there is a poly-bounded $R \in \mathbf{P}$ such that $\forall x$

$$x \in L \iff \exists w_1 \forall w_2 \exists w_3 \ldots Q w_k \ \langle x, w_1, w_2, \ldots, w_k \rangle \in R,$$

where $Q = \exists$ if k is odd
and $Q = \forall$ if k is even.

Let us prove the equivalence!

The proof of equivalence — induction on k

Base: for $k = 1$ this is the definition of $\Sigma_1^p = \mathbf{NP}$ (and $\Pi_1^p = \mathbf{co}\text{-}\mathbf{NP}$)
($L \in \mathbf{NP} \iff$ there is $T \in \mathbf{P}$ s.t. $L = \{x : \exists w \ \langle x, w\rangle \in T\}$).
($L \in \mathbf{co}\text{-}\mathbf{NP} \iff$ there is $S \in \mathbf{P}$ s.t. $L = \{x : \forall w \ \langle x, w\rangle \in S\}$).

The proof of equivalence — induction on k

Step $(k-1 \mapsto k)$:

$L \in \Sigma_k^p = \mathbf{NP}^{\Sigma_{k-1}^p} \Rightarrow$ there is an oracle p-time NTM $N^\bullet$ and an oracle $O \in \Sigma_{k-1}^p$ s.t. $L = L(N^O)$.

Induction hypothesis $\Longrightarrow$ $O = \{q : \exists w \ \langle q, w \rangle \in T'\}$, where $T' \in \Pi_{k-2}^p$.

The proof of equivalence — induction on k

Step $(k-1 \mapsto k)$:
$L \in \Sigma_k^p = \mathbf{NP}^{\Sigma_{k-1}^p} \Rightarrow$ there is an oracle p-time NTM $N^\bullet$ and an oracle $O \in \Sigma_{k-1}^p$ s.t. $L = L(N^O)$.

Induction hypothesis $\Longrightarrow$ $O = \{q : \exists w \ \langle q, w \rangle \in T'\}$, where $T' \in \Pi_{k-2}^p$.

To decide $x \in^? L$ using a lower class, we need to run N^O, but how? We do not have O.

Thus let us ask for a witness $(\exists w_1)$ and check it:

(1) The accepting path of N^O [see the whiteboard].

(2) The answers $a_1, a_2, \ldots, a_t$ of the oracle O for each query q_t of N^O.

Checking (1) is easy if we have a_i's (run UTM).
Checking (2) without O seems impossible unless we have more info in the witness.

The proof of equivalence — induction on k

Step $(k-1 \mapsto k)$:
$L \in \Sigma_k^p = \mathbf{NP}^{\Sigma_{k-1}^p} \Rightarrow$ there is an oracle p-time NTM $N^\bullet$ and an oracle $O \in \Sigma_{k-1}^p$ s.t. $L = L(N^O)$.

Induction hypothesis $\Longrightarrow$ $O = \{q : \exists \boldsymbol{w}\ \langle q, w\rangle \in T'\}$, where $T' \in \Pi_{k-2}^p$.

To decide $x \in^? L$ using a lower class, we need to run N^O, but how? We do not have O.

Thus let us ask for a witness $(\exists w_1)$ and check it:

(1) The accepting path of N^O [see the whiteboard].

(2) The answers $a_1, a_2, \ldots, a_t$ of the oracle O for each query q_t of N^O.

Checking (1) is easy if we have a_i's (run UTM).
Checking (2) without O seems impossible unless we have more info in the witness.

(3) For each t s.t. $a_t = 1$ the witness includes $\boldsymbol{w} = w^{(t)}$ for $q_t \in O$ (see $\uparrow$).

It remains to check (1) using UTM and check (2) using (3) for $a_t = 1$.

Checking the witness using $k-1$ quantifiers

"Program" for $T \in \Pi^p_{k-1}$ s.t. $L = \{x : \exists w_1 \ \langle x, w_1 \rangle \in T\}$:

1. Check (1) using UTM and a_i's. // *0 quantifiers*
2. For every $a_t = 1$, check this answer using $w^{(t)}$ and T': $\langle q_t, w^{(t)}, w_3, \ldots \rangle \in T'$. // $T' \in \Pi^p_{k-2}$
3. For every $a_t = 0$, check this answer using $\overline{O}$. // $\overline{O} \in \Pi^p_{k-1}$

How to merge many Π^p_{k-1} computations into one?

Version 1 (using logic): Apply the induction hypothesis iteratively:

$\overline{O} = \{q : \forall w_2 \exists w_3 \forall w_4 \ldots Q w_k \ \langle q, w_2, w_3, \ldots, w_k \rangle \in \overline{S}\}$,

Observe that for different queries q_t the variables are disjoint.

Recall (logic!) that $\forall u \ A(q,u) \ \wedge \ \forall v \ B(q,v) \quad \equiv \quad \forall u \forall v \ (A(q,u) \wedge B(q,v))$.

Thus take the conjunction of all the formulas:
$\forall w_2^{(1)}, w_2^{(2)}, \ldots \exists w_3^{(1)}, w_3^{(2)}, \ldots$ [p-time machine verifying that all the checks are OK].

Version 2 (without logic): [see whiteboard].

Complete problems for the classes of the Polynomial Hierarchy

Complete problems for Σ_k^p

Theorem

$\texttt{QBF}_k = \{$ True formulas of type $\exists w_1 \forall w_2 \ldots Q w_k \ \Phi(x, w_1, \ldots, w_k)\}$ is Σ_k^p-complete,

What is Φ? First use a circuit, then use CNF/DNF.

$L \in \Sigma_k^p$ (and let k be odd)
$\implies L = \{x : \exists w_1 \forall w_2 \ldots \exists w_k \ \langle x, w_1, \ldots, w_k\rangle \in R\}$, where $R \in \mathbf{P}$ (time $p(n)$).
$\implies L = \{x : \exists w_1 \forall w_2 \ldots \exists w_k \ C_n(\langle x, w_1, \ldots, w_k\rangle) = 1\}$, where C_n is a circuit for R and $n = |x|$.

Reduction r from L to $\texttt{QBF}_k$ (almost):
On input x_* of length n,

- build C_n for the $p(n)$-time run of R on x (recall the **NP**-completeness of `Circuit-SAT`),
- Output $\exists w_1 \forall w_2 \ldots \exists w_k \ C_n|_{x=x^*}(\langle w_1, \ldots, w_k\rangle)$ (that is, we substitute value for the input bits of x in $C_n(\langle x, w_1, \ldots, w_k\rangle)$, and $w_1, \ldots, w_k$ remain variables).

Recall the reduction `Circuit-SAT` $\to$ `3-SAT`:
$f(C_n) = \Phi$ s.t. $\exists a \ C_n(a) = 1 \iff \exists a \ \exists b \ \Phi(a, b) = 1$.

Thus output quantified Φ instead of quantified $C_n(\ldots)$ (for $a = \langle w_1, \ldots, w_k\rangle$), the quantifiers $\exists w_k \exists b$ merge.

Complete problems for Σ_k^p

Theorem

$\texttt{QBF}_k = \{$ True formulas of type $\exists w_1 \forall w_2 \ldots Q w_k \ \Phi(x, w_1, \ldots, w_k)\}$ is Σ_k^p-complete,
where Φ is in 3-CNF if $Q = \exists$, and Φ is in 3-DNF if $Q = \forall$.

What is Φ? First use a circuit, then use CNF/DNF.

$L \in \Sigma_k^p$ (and let k be odd)
$\implies L = \{x : \exists w_1 \forall w_2 \ldots \exists w_k \ \langle x, w_1, \ldots, w_k\rangle \in R\}$, where $R \in \mathbf{P}$ (time $p(n)$).
$\implies L = \{x : \exists w_1 \forall w_2 \ldots \exists w_k \ C_n(\langle x, w_1, \ldots, w_k\rangle) = 1\}$, where C_n is a circuit for R and $n = |x|$.

Reduction r from L to $\texttt{QBF}_k$ (almost):
On input x_* of length n,

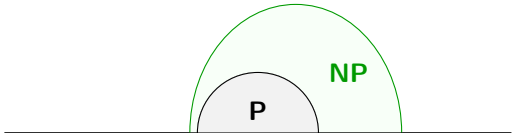
- build C_n for the $p(n)$-time run of R on x (recall the **NP**-completeness of `Circuit-SAT`),
- Output $\exists w_1 \forall w_2 \ldots \exists w_k \ C_n|_{x=x^*}(\langle w_1, \ldots, w_k\rangle)$ (that is, we substitute value for the input bits of x in $C_n(\langle x, w_1, \ldots, w_k\rangle)$, and $w_1, \ldots, w_k$ remain variables).

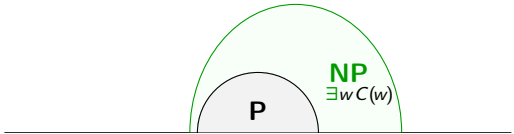
Recall the reduction `Circuit-SAT` $\to$ `3-SAT`:
$f(C_n) = \Phi$ s.t. $\exists a \ C_n(a) = 1 \iff \exists a \ \exists b \ \Phi(a, b) = 1$.

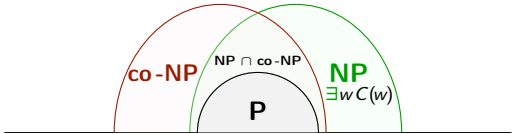
Thus output quantified Φ instead of quantified $C_n(\ldots)$ (for $a = \langle w_1, \ldots, w_k\rangle$), the quantifiers $\exists w_k \exists b$ merge.

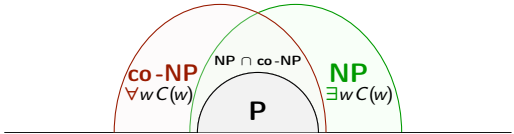
If k is even: $\overline{L} = \{x : \forall w_1 \exists w_2 \ldots \exists w_k \ \langle x, w_1, \ldots, w_k\rangle \in \overline{R}\}$, convert $\overline{R}$ to C_n, to a CNF, negate... it's DNF!

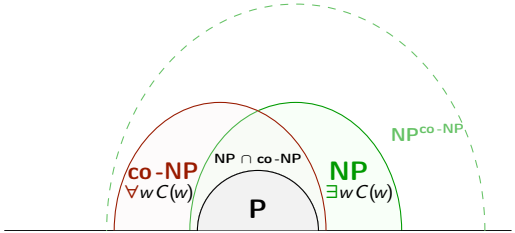

P

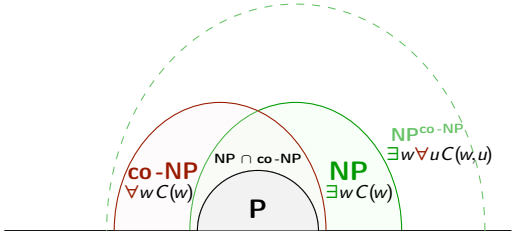

NP
P

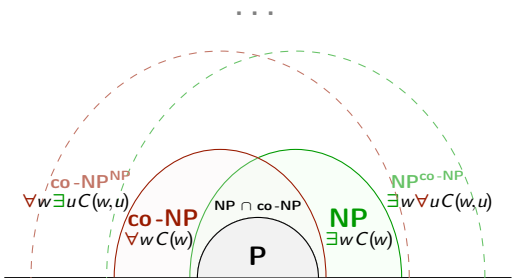

NP
$\exists w\, C(w)$
P

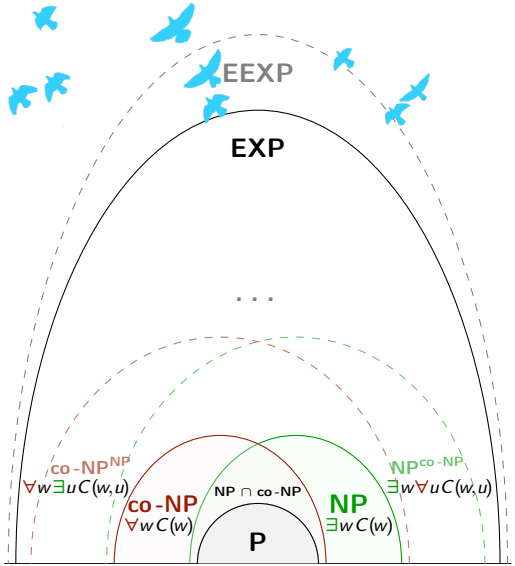

co-NP
NP ∩ co-NP
NP
∃w C(w)
P

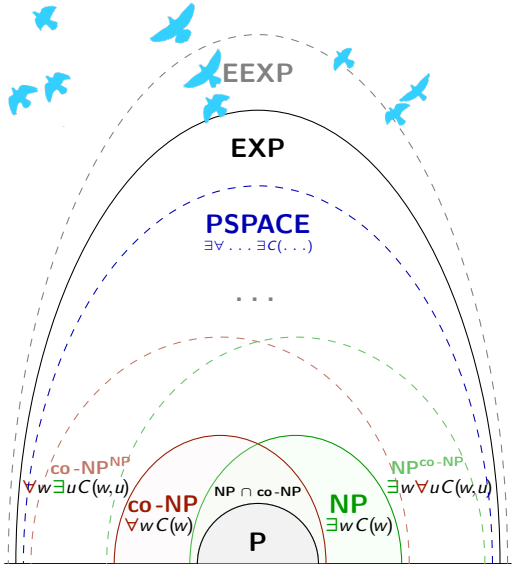

co-NP
$\forall w\, C(w)$
NP ∩ co-NP
P
NP
$\exists w\, C(w)$


co-NP
$\forall w\, C(w)$
NP ∩ co-NP
NP
$\exists w\, C(w)$
P
$NP^{co\text{-}NP}$


co-NP
$\forall w\, C(w)$
NP ∩ co-NP
NP
$\exists w\, C(w)$
P
$\mathbf{NP}^{\mathbf{co\text{-}NP}}$
$\exists w \forall u\, C(w,u)$


...
co-NP^NP
∀w∃u C(w,u)
co-NP
∀w C(w)
NP ∩ co-NP
NP
∃w C(w)
NP^co-NP
∃w∀u C(w,u)
P


EEXP
EXP
. . .
co-NP^NP
∀w∃u C(w,u)
co-NP
∀w C(w)
NP ∩ co-NP
P
NP
∃w C(w)
NP^co-NP
∃w∀u C(w,u)


EEXP
EXP
PSPACE
$\exists\forall\ldots\exists C(\ldots)$
$\cdots$
co-NP^{NP}
$\forall w \exists u C(w,u)$
co-NP
$\forall w C(w)$
NP ∩ co-NP
P
NP
$\exists w C(w)$
$\text{NP}^{\text{co-NP}}$
$\exists w \forall u C(w,u)$

Collapsing the Polynomial Hierarchy

Easy observations

Observation 0. All these classes are closed under many-one p-time reductions.
(Incorporate the reduction into the main machine).
If $L \to^p L'$ and $L' \in \Sigma_k^p$, then $L \in \Sigma_k^p$.

Observation 1. If $\mathbf{P} = \mathbf{NP}$, then all these classes coincide.

Easy observations

Observation 0. All these classes are closed under many-one p-time reductions.
(Incorporate the reduction into the main machine).
If $L \to^p L'$ and $L' \in \Sigma_k^p$, then $L \in \Sigma_k^p$.

Observation 1. If $\mathbf{P} = \mathbf{NP}$, then all these classes coincide.

Observation 2. If $\Sigma_k^p = \Pi_k^p$, then $\Sigma_k = \Sigma_{k+1} = \ldots$, etc.
(Because negating $T \in \Pi_k^p$ in $\exists w_1\ \langle x, w_1 \rangle \in T$ ($\equiv \exists w_1 \forall w_2 \ldots$) will change the second quantifier to $\exists$.)

This is called the collapse of the hierarchy. We do not believe in it.

Observation 3. $\mathbf{PH} = \bigcup_{k \in \mathbb{N}} \Sigma_k^p$ has no complete languages unless it collapses.

Proof: if L' is complete for **PH**
$\Rightarrow$ $L' \in \Sigma_k^p$ for some k
$\Rightarrow$ every $L \in \mathbf{PH}$ reduces to L', thus $L \in \Sigma_k^p$ as well
$\Rightarrow$ $\mathbf{PH} \subseteq \Sigma_k^p$.

Polynomial-size circuits and the Karp–Lipton theorem

Polynomial-size circuits

Let us replace a poly-time DTM with poly-size circuits.

Definition (Size)

$L \in \mathbf{Size}[f(n)] \iff$ there is a family $\{C_n\}_{n\in\mathbb{N}}$ of circuits (n is the number of inputs) of size $O(f(n))$ s.t. $\forall x \in \{0,1\}^*$

$$x \in L \iff C_{|x|}(x) = 1.$$

Definition (P/poly)

$$\mathbf{P/poly} = \bigcup_{k\geqslant 1} \mathbf{Size}[n^k].$$

Theorem

$\mathbf{P} \subseteq \mathbf{P/poly}$.

(Recall the **NP**-completeness of `Circuit-SAT`:
We can simulate a DTM using a circuit if n and the running time are known.)

Observation: $\mathbf{P/poly} \not\subseteq \mathbf{P}$.
(**P/poly** contains undecidable languages!) (E.g., $\{x : \text{DTM with Turing number } |x| \text{ accepts } \varepsilon\}$.)

The Karp–Lipton theorem

Polynomial-size circuits are much stronger than p-time DTMs.
Can they solve SAT? Unlikely...

Theorem (R. Karp, R. Lipton, M. Sipser)

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

Remark

There are much deeper collapses! Although not yet to Δ_2^p.

To be continued...