Today: Polynomial Hierarchy & P/poly

Polynomial Hierarchy

We know $P \subseteq NP \subseteq PSPACE$

Informally, how much "room" is there between NP & PSPACE?

Are most natural problems that seem to be outside NP (but in PSPACE) PSPACE-complete?

Eg: Exact-INDSET = { <G, k> | the largest independent set in G has size exactly k }

Exact-INDSET is unlikely to be in NP or co-NP.

Another reason to find classes between NP & PSPACE

Complete problem for NP: SAT

$\exists x_1 \exists x_2 \cdots \exists x_n \; F(x_1, \ldots, x_n)$

Complete problem for co-NP: UNSAT

$\forall x_1 \forall x_2 \cdots \forall x_n \neg F(x_1, \ldots, x_n)$

Complete problem for PSPACE: TQBF

$Q_1 x_1 \, Q_2 x_2 \cdots Q_n x_n \; F(x_1, \ldots, x_n)$

Q_i is either $\exists$ or $\forall$.

switching between $\exists$ & $\forall$.

It seems like alternations add power!

Why not try fewer alternations (1, 2 etc.) ?

Σ_i^P :

$L \in \Sigma_i^P$ if and only if there is

a poly time DTM M s.t. for any $x \in \{0,1\}^*$

$x \in L \iff \exists y_1 \forall y_2 \exists y_3 \cdots Q_i y_i$

$M(x, y_1, y_2, \ldots, y_i) = 1$

where $y_1, \ldots, y_i$ are Boolean strings

of length $\leq poly(|x|)$

$\Sigma_1^P = NP$, Σ_i^P - $\underline{(i-1)}$ alternations

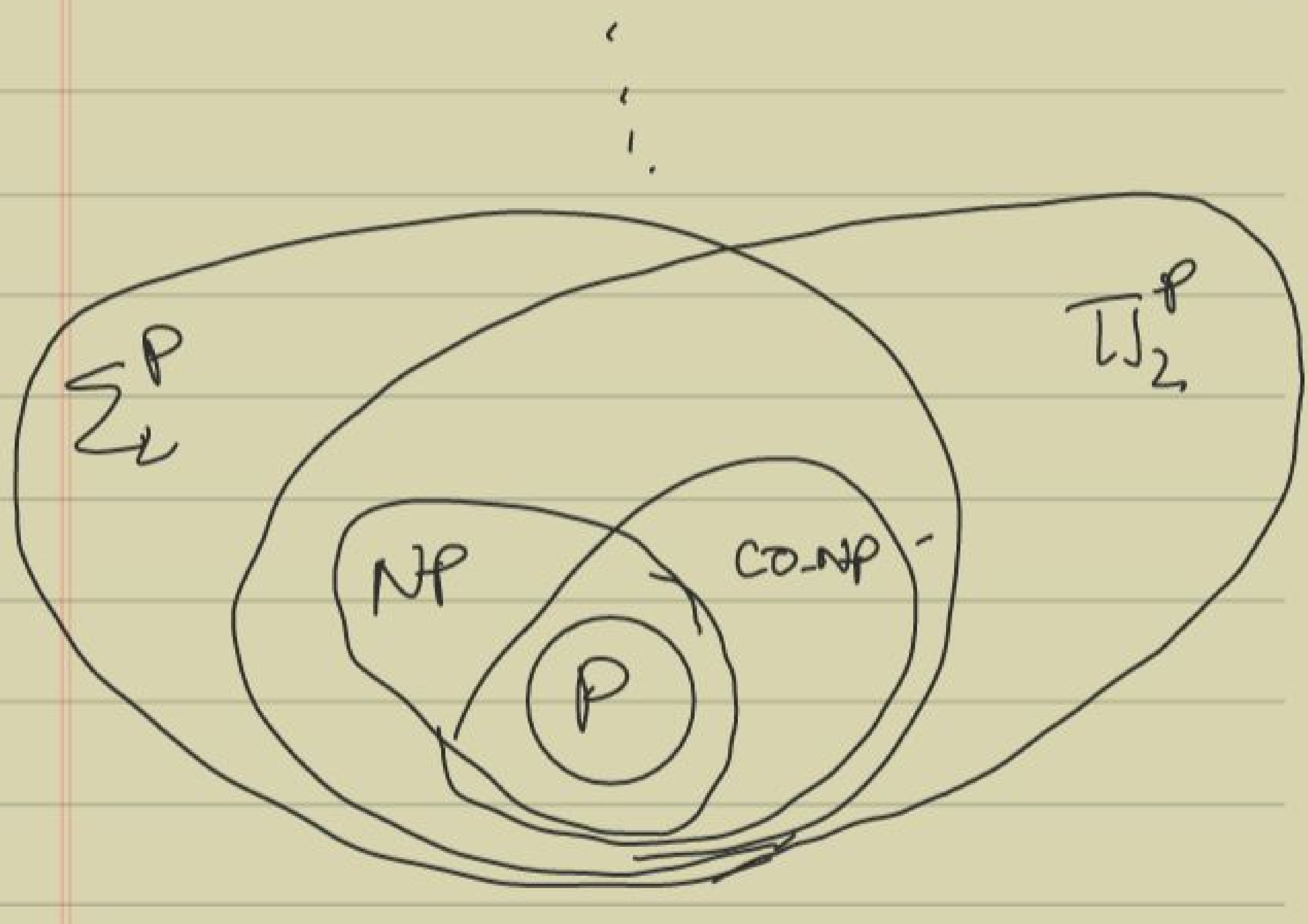
$\Pi_i^P = co\text{-}\Sigma_i^P = \{\bar{L} \mid L \in \Sigma_i^P\}$

Can also be defined using i quantifiers

starting with $\forall$.

Obs: $\Sigma_i^P \subseteq \Sigma_{i+1}^P \cap \Pi_{i+1}^P$

$\Pi_i^P \subseteq \Sigma_{i+1}^P \cap \Pi_{i+1}^P$



$$PH = \bigcup_{i=1}^{\infty} \Sigma_i^P = \bigcup_{i=1}^{\infty} \Pi_i^P$$

Is $\Sigma_i^P = \Sigma_{i+1}^P$? (Higher analogue of P vs. NP)

Is $\Sigma_i^P = \Pi_i^P$? (Higher analogue of NP vs co-NP)

Expected answers: NO (for same to both reasons)

This is implied by the assumption that the Polynomial Hierarchy is infinite.

i.e $PH \neq \Sigma_i^P$ for any i.

(aka "Polynomial Hierarchy does not collapse")

Standard assumption in Complexity theory.

Strong version of $P \neq NP$

Thm 1: ① If $\Sigma_i^P = \Sigma_{i+1}^P$ for any i,

then $PH = \Sigma_i^P$

("PH collapses to i^{th} level")

② If $\Sigma_i^P = \Pi_i^P$ for any i, then

$PH = \Sigma_i^P$.

Pf of ② (① is similar).

Let's do it for $i=1$ (same for layer i)

Assume $\Sigma_1^P = \Pi_1^P$. Now we will

show $\Sigma_2^P = \Sigma_1^P$.

We already know: $\Sigma_1^P \subseteq \Sigma_2^P$. So

we only need $\Sigma_2^P \subseteq \Sigma_1^P$.

Fix any $L \in \Sigma_2^P$. To show: $L \in \Sigma_1^P$.

$L \in \Sigma_2^P \Rightarrow$ Polytime DTM M s.t. for any x

$$x \in L \iff \underbrace{\exists y_1}\ \underbrace{\forall y_2}_{\text{length} \leq P(|x|)} M(x, y_1, y_2) = 1$$

Define $L' = \{(x, y_1) \mid |y_1| \leq P(|x|)$
$\forall y_2\ M(x, y_1, y_2) = 1\}$

Obs: (1) $L = \{x \mid \exists y_1\ (x, y_1) \in L'\}$

(2) $L' \in \Pi_1^P \hookrightarrow \boxed{L = \exists \cdot L'}$

By assumption $\Pi_1^P = \Sigma_1^P$ & thus
$L' \in \Sigma_1^P$

$\Rightarrow$ There is a polytime M' s.t.

$(x, y_1) \in L' \iff \exists y_2'\ M'(x, y_1, y_2') = 1$

Thus,

$$x \in L \iff \exists y_1\ (x, y_1) \in L'$$

$$\iff \underbrace{\exists y_1 \exists y_2'}\ M'(x, y_1, y_2') = 1$$

$$\iff \exists y_1'\ M'(x, y_1') = 1$$

$$|y_1'| \leq |y_1| + |y_2'| \leq poly(|x|)$$

Hence we have shown $L \in \Sigma_1^P$!

So $\Sigma_2^P = \Sigma_1^P$ & hence

$$\Pi_2^P = co\text{-}\Sigma_2^P = co\text{-}\Sigma_1^P = \underbrace{\Pi_1^P = \Sigma_1^P}_{\text{by assumption!}}$$

$\Rightarrow \Sigma_1^P = \Sigma_2^P = \Pi_2^P$. Continuing the argument, $PH = \Sigma_1^P$. $\boxed{\Xi}$

Complete problems

$\exists$ if i odd
$\forall$ if i even

vector of Boolean vars

Σ_i-SAT =

$\{\exists y_1 \forall y_2 \cdots Q_i y_i\, F(y_1, \ldots, y_i) \mid$

the TQBF evaluates to 1$\}$

Π_i-SAT =

$\{\forall y_1 \exists y_2 \cdots Q_i y_i\, F(y_1, \ldots, y_i) \mid$

the TQBF evaluates to 1$\}$

Thm 2: Σ_iSAT is Σ_i^P-complete &

Π_i SAT is Π_i^P-complete

w.r.t. polynomial-time reductions

[also true under log space reductions]

What about complete problems for PH?

Thm: PH does not have complete problems unless it collapses.

Proof: Say $L \in PH$ is PH-complete.

Then $L \in \Sigma_i^P$ for some fixed i.

We will show: $PH = \Sigma_i^P$.

Already know: $\Sigma_i^P \subseteq PH$.

Need to show: $PH \subseteq \Sigma_i^P$.

Fix any $L' \in PH$. We know $L' \leq_p L$.

Let f be a reduction from L' to L.

Then

$x \in L' \iff f(x) \in L$

$\iff \underbrace{\exists y_1 \forall y_2 \cdots Q_i y_i \; M(f(x), y_1, \ldots, y_i) = 1}_{\text{using the fact that } L \in \Sigma_i^P}$

$\iff \underbrace{\exists y_1 \forall y_2 \cdots Q_i y_i \; M'(x, y_1, \ldots, y_i) = 1}_{M' \text{ simply runs } f \text{ on } x \text{ \& then applies } M \text{ on } (f(x), y_1, \ldots, y_i)}$

This show $L' \in \Sigma_i^P$!

Hence, $PH \subseteq \Sigma_i^P$. $\square$

Corollary 3: PH $\neq$ PSPACE unless PH collapses.

Equivalently: PH = PSPACE $\Rightarrow$ PH collapses.

Proof: If PH = PSPACE, TQBF is PH-complete. Now use Thm. 2

Two more definitions of PH:

→ Oracle TMs

→ Alternating TMs. (skipped, see textbook)

Oracle TMs:

$\mathcal{C}$ - a complexity class ($P, NP, \Sigma_i^P, \Pi_i^P$ etc.)

$NP^{\mathcal{C}}$ - languages decided by poly. time NTMs with oracle access to some language $L \in \mathcal{C}$.

Alternate characterization of PH:

$$\Sigma_2^P = NP^{NP} = NP^{SAT}$$

More generally, $\Sigma_i^P = NP^{\Sigma_{i-1}^P}$
$(i \geq 2)$ $= NP^{\Sigma_{i-1}SAT}$

[Proof in textbook, but we'll skip it.]

Counting class #P

Another generalization of NP & co-NP.

$L \in NP$ (coNP) if & only if there is a poly-time DTM M s.t.

$$x \in L \iff \exists y \ (\forall y) \quad |y| \leq poly(|x|)$$
$$M(x, y) = 1$$

What if we could count the number of "certificates", i.e the number of y s.t. $M(x, y) = 1$?

Then we could solve both NP-complete & co-NP-complete problems!

Let M be a poly-time DTM.
that accepts or rejects inputs of the form (x, y) where $|y| = p(|x|)$ a polynomial.

We define $\#_M : \{0,1\}^* \to \mathbb{N}$ as

$$\#_M(x) = \left|\left\{ y \in \{0,1\}^{p(|x|)} \mid \begin{matrix} M(x,y) \\ = 1 \end{matrix} \right\}\right|$$

<u>Note</u>: $0 \leq \#_M(x) \leq 2^{p(|x|)}$ & so
<u>can be expressed using $p(|x|)$ many bits.</u>

$\#P$ = Set of all such $\#_M$.

$\hookrightarrow$ a class of <u>function</u> problems, <u>not</u> decision problems.

Examples:

① #SAT -

$\#SAT(\varphi)$ = Number of satisfy-
-ing assignments.

CNF formula

② #BIPARTITE-MATCHING (or #BM)

$\#BM(G)$ = Number of perfect
matchings in G.

bipartite graph

③ #SPANNING-TREE (or #ST)

$\#ST(G)$ = Number of spanning
Trees in G.

undirected graph

#P & PH

$P^{\#P}$ = decision problems solvable by a poly-time DTM with an oracle for a function in #P.

Clearly, with a #SAT oracle we can solve all problems in NP & co-NP.

But we can go much further...

Toda's thm: $P^{\#P} \supseteq PH$.

Counting is at least as strong as the polynomial hierarchy!

Completeness

$FP = \{ f : \{0,1\}^* \to \mathbb{N} \mid f$ computed by a poly-time DTM $\}$

f is #P-complete if:

① $f \in \#P$ (i.e $f = \#_M$ for some M)

② For any $g \in \#P$, $g \in FP^f$

#SAT is #P-complete (Careful analysis of Cook-Levin)

Valiant's: #BM is #P-complete.

thus

Surprising because the decision version of bipartite matching is easy!