Computational Complexity

Lecture 8
More of the Polynomial Hierarchy
Oracle-based Definition
Recall PH

\[ \{ x \mid \exists u_1 \forall u_2 \exists u_3 \ F(x,u_1,u_2,u_3) \} \]

\[ \{ x \mid \forall u_1 \exists u_2 \forall u_3 \ F(x,u_1,u_2,u_3) \} \]

\[ \{ x \mid \exists u_1 \forall u_2 \ F(x,u_1,u_2) \} \]

\[ \{ x \mid \forall u_1 \exists u_2 \ F(x,u_1,u_2) \} \]

\[ \{ x \mid \exists u_1 \ F(x,u_1) \} \]

\[ \{ x \mid \forall u_1 \ F(x,u_1) \} \]

\[ \{ x \mid F(x) \} \]
Oracle Machines
Oracle Machines

Recall Oracle Machine
Oracle Machines

- Recall Oracle Machine
  - Writes queries on query-tape, enters and leaves query state, and expects answer from oracle on the tape
Oracle Machines

- Recall Oracle Machine
  - Writes queries on query-tape, enters and leaves query state, and expects answer from oracle on the tape
  - Can run an oracle machine with any oracle
Oracle Machines

- Recall Oracle Machine
  - Writes queries on query-tape, enters and leaves query state, and expects answer from oracle on the tape
  - Can run an oracle machine with any oracle
  - Oracle fully specified by the input-output behavior
Oracle Machines

Recall Oracle Machine

- Writes queries on query-tape, enters and leaves query state, and expects answer from oracle on the tape
- Can run an oracle machine with any oracle
- Oracle fully specified by the input-output behavior
- Language oracle: answer is a single bit
Oracle Machines

Recall Oracle Machine

- Writes queries on query-tape, enters and leaves query state, and expects answer from oracle on the tape
- Can run an oracle machine with any oracle
- Oracle fully specified by the input-output behavior
- Language oracle: answer is a single bit
- This is what we consider
Oracle Machines (ctd.)
Oracle Machines (ctd.)

- Non-deterministic oracle machine
Oracle Machines (ctd.)

- Non-deterministic oracle machine
  - Can make non-deterministic choices and make oracle queries. (Note: oracles are deterministic!)
Oracle Machines (ctd.)

- Non-deterministic oracle machine
- Can make non-deterministic choices and make oracle queries. (Note: oracles are deterministic!)
- Said to accept if any thread reaches accept state
Oracle Machines (ctd.)

- Non-deterministic oracle machine
  - Can make non-deterministic choices and make oracle queries. (Note: oracles are deterministic!)
  - Said to accept if any thread reaches accept state
  - Equivalently, a deterministic oracle machine which takes a (read-once) certificate $w$ (the list of non-deterministic choices)
Oracle Machines (ctd.)

- Non-deterministic oracle machine
  - Can make non-deterministic choices and make oracle queries. (Note: oracles are deterministic!)
  - Said to accept if any thread reaches accept state
  - Equivalently, a deterministic oracle machine which takes a (read-once) certificate \( w \) (the list of non-deterministic choices)
  - Said to accept \( x \) if \( \exists w \) such that \( (x,w) \) takes it to accepting state
Oracle Machines (ctd.)

- **Non-deterministic oracle machine**
  - Can make non-deterministic choices and make oracle queries. (Note: oracles are deterministic!)
  - Said to accept if any thread reaches accept state
  - Equivalently, a deterministic oracle machine which takes a (read-once) certificate $w$ (the list of non-deterministic choices)
    - Said to accept $x$ if there exists $w$ such that $(x,w)$ takes it to accepting state
$\text{NP}^A$

$\text{NP}^A$: class of languages accepted by oracle NTMs with oracle for $A$ in poly time
NP^A

NP^A: class of languages accepted by oracle NTMs with oracle for A in poly time

Certificate version: NP^A has languages of the form
$\text{NP}^A$

- $\text{NP}^A$: class of languages accepted by oracle NTMs with oracle for $A$ in poly time

- Certificate version: $\text{NP}^A$ has languages of the form
  \[ B = \{ x \mid \exists w \ M^A(x,w) = 1 \} \]
$NP^A$

- $NP^A$: class of languages accepted by oracle NTMs with oracle for $A$ in poly time
- Certificate version: $NP^A$ has languages of the form $B = \{x \mid \exists w \ M^A(x,w) = 1\}$
- where $M$ deterministic oracle machine
\textbf{NP}^A

- \textbf{NP}^A: class of languages accepted by oracle NTMs with oracle for \( A \) in poly time

- Certificate version: \( \text{NP}^A \) has languages of the form

\[ B = \{ x \mid \exists w \ M^A(x, w) = 1 \} \]

- where \( M \) deterministic oracle machine

- \( M^A \) runs in poly(|x|) time and |w|=poly(|x|)
$\text{NP}^A$

- $\text{NP}^A$: class of languages accepted by oracle NTMs with oracle for $A$ in poly time

- Certificate version: $\text{NP}^A$ has languages of the form

  $B = \{x \mid \exists w \ M^A(x,w) = 1\}$

  where $M$ deterministic oracle machine

  $M^A$ runs in $\text{poly}(|x|)$ time and $|w|=\text{poly}(|x|)$

- $\text{co-}(\text{NP}^A) = (\text{co-NP})^A$
\( \text{NP}^A \)

\( \text{NP}^A \): class of languages accepted by oracle NTMs with oracle for A in poly time

Certificate version: \( \text{NP}^A \) has languages of the form

\[ B = \{x \mid \exists w \ M^A(x,w) = 1\} \]

where \( M \) deterministic oracle machine

\( M^A \) runs in poly(|x|) time and |w|=poly(|x|)

\( \text{co-(NP}^A\) = (co-NP)^A

languages of the form \( \{x \mid \forall w \ M^A(x,w) = 1\} \)
NPA
If $A$ in $P$, $NPA = NP$
$\text{If } A \text{ in } P, \text{ } \mathbf{NP}^A = \mathbf{NP}$

- Can “implement” the oracle as a subroutine
\( \text{NP}^A \)

- If \( A \) in \( P \), \( \text{NP}^A = \text{NP} \)
- Can “implement” the oracle as a subroutine
- If \( A \) in \( \text{NP} \)?
If $A$ in P, $NPA = NP$

Can "implement" the oracle as a subroutine

If $A$ in NP?

Oracle for $A$ is an oracle for $A^c$ too! $NPA = NPA^c$
$NP^A$

- If $A$ in $P$, $NP^A = NP$
  - Can “implement” the oracle as a subroutine
- If $A$ in $NP$?
  - Oracle for $A$ is an oracle for $A^c$ too! $NP^A = NP^{A^c}$
  - $NP \cup co-NP \subseteq NP^{SAT}$
If $A$ in $P$, $NP^A = NP$

Can “implement” the oracle as a subroutine

If $A$ in $NP$?

Oracle for $A$ is an oracle for $A^c$ too! $NP^A = NP^{A^c}$

$NP \cup \text{co-NP} \subseteq NP^{SAT}$

Can we better characterize $NP^{SAT}$?
NP and relatives
\[ \text{NP}^{\text{NP}} \text{ and relatives} \]

\[ \text{NP}^{\text{SAT}} = \bigcup_{A \in \text{NP}} \text{NP}^A \]
$\text{NP}^\text{NP}$ and relatives

$\text{NP}^{\text{SAT}} = \bigcup_{A \in \text{NP}} \text{NP}^A$

Oracle for A can be implemented using oracle for SAT in polynomial time (deterministically)
NP^NP and relatives

\[ \text{NP}^\text{SAT} = \bigcup_{A \in \text{NP}} \text{NP}^A \]

- Oracle for $A$ can be implemented using oracle for SAT in polynomial time (deterministically)
- $\text{NP}^\text{SAT}$ also called $\text{NP}^{\text{NP}}$
$\mathbf{NP}^{\mathbf{NP}}$ and relatives

- $\mathbf{NP}^{\mathbf{SAT}} = \bigcup_{A \in \mathbf{NP}} \mathbf{NP}^A$

- Oracle for $A$ can be implemented using oracle for SAT in polynomial time (deterministically)

- $\mathbf{NP}^{\mathbf{SAT}}$ also called $\mathbf{NP}^{\mathbf{NP}}$

- $\mathbf{NP}^{\Sigma_k} = \bigcup_{A \in \Sigma_k} \mathbf{NP}^A = \mathbf{NP}^{\Sigma_k\mathbf{SAT}}$
NP^{NP} and relatives

- **NP^{SAT} = \bigcup_{A \in NP} NP^A**
  - Oracle for A can be implemented using oracle for SAT in polynomial time (deterministically)

- **NP^{SAT} also called NP^{NP}**

- **NP^{Σ_k} = \bigcup_{A \in Σ_k} NP^A = NP^{Σ_kSAT}**

- Will show **NP^{Σ_k} = Σ_{k+1}^P** (alt. definition for Σ_{k+1}^P)
$\text{NP}^{\text{NP}}$ and relatives

- $\text{NP}^{\text{SAT}} = \bigcup_{A \in \text{NP}} \text{NP}^A$

  - Oracle for $A$ can be implemented using oracle for SAT in polynomial time (deterministically)

- $\text{NP}^{\text{SAT}}$ also called $\text{NP}^{\text{NP}}$

- $\text{NP}^{\Sigma_k} = \bigcup_{A \in \Sigma_k} \text{NP}^A = \text{NP}^{\Sigma_k \text{SAT}}$

  - Will show $\text{NP}^{\Sigma_k} = \Sigma_{k+1}^P$ (alt. definition for $\Sigma_{k+1}^P$)

- In particular, $\text{NP}^{\text{NP}} = \Sigma_2^P$
$\mathbf{NP}^{\mathbf{NP}}$ and relatives

1. $\mathbf{NP}^{\mathbf{SAT}} = \bigcup_{A \in \mathbf{NP}} \mathbf{NP}^A$

   - Oracle for $A$ can be implemented using oracle for SAT in polynomial time (deterministically)

2. $\mathbf{NP}^{\mathbf{SAT}}$ also called $\mathbf{NP}^{\mathbf{NP}}$

3. $\mathbf{NP}^{\Sigma_k} = \bigcup_{A \in \Sigma_k} \mathbf{NP}^A = \mathbf{NP}^{\Sigma_k \mathbf{SAT}}$

   - Will show $\mathbf{NP}^{\Sigma_k} = \Sigma_{k+1}^P$ (alt. definition for $\Sigma_{k+1}^P$)

   - In particular, $\mathbf{NP}^{\mathbf{NP}} = \Sigma_2^P$
$\Sigma_{k+1} = \mathsf{NP}^{\Sigma_k}$
$\Sigma_{k+1} = \text{NP}^{\Sigma_k}$

Consider $L \in \Sigma_{k+1}^P$. 
\[ \Sigma_{k+1} = \mathsf{NP}^{\Sigma_k} \]

Consider \( L \in \Sigma_{k+1}^p \)

\( L = \{ x \mid \exists w \ (x,w) \in L' \} \), where \( L' \) in \( \Pi_k^p \)
\[ \Sigma_{k+1} = \text{NP}^{\Sigma_k} \]

Consider \( L \in \Sigma_{k+1}^P \)

\( L = \{ x | \exists w (x, w) \in L' \} \), where \( L' \) in \( \Pi_k^P \)

So \( L \) in \( \text{NP}^{L'} \) where \( L' \) in \( \Pi_k^P \)
$\Sigma_{k+1} = NP^{\Sigma_k}$

Consider $L \in \Sigma_{k+1}^P$

$L = \{ x | \exists w \ (x,w) \in L' \}, \text{ where } L' \in \Pi_k^P$

So $L \in NP^{L'}$ where $L' \in \Pi_k^P$

But $NP^{L'} \subseteq NP^{\Pi_k} = NP^{\Sigma_k}$
\[ \Sigma_{k+1} = \text{NP}^{\Sigma_k} \]

- Consider \( L \in \Sigma_{k+1}^P \)
  - \( L = \{ x | \exists w \ (x,w) \in L' \} \), where \( L' \) in \( \Pi_k^P \)
  - So \( L \) in \( \text{NP}^{L'} \) where \( L' \) in \( \Pi_k^P \)
    - But \( \text{NP}^{L'} \subseteq \text{NP}^{\Pi_k} = \text{NP}^{\Sigma_k} \)
  - So \( \Sigma_{k+1}^P \subseteq \text{NP}^{\Sigma_k} \)
\[ \Sigma_{k+1} = NP^{\Sigma_k} \]

Consider \( L \in \Sigma_{k+1}^P \)

\( L = \{ x \mid \exists w \ (x,w) \in L' \}, \) where \( L' \) in \( \Pi_k^P \)

So \( L \) in \( NP^{L'} \) where \( L' \) in \( \Pi_k^P \)

But \( NP^{L'} \subseteq NP^{\Pi_k} = NP^{\Sigma_k} \)

So \( \Sigma_{k+1}^P \subseteq NP^{\Sigma_k} \)

Now to show \( NP^{\Sigma_k} \subseteq \Sigma_{k+1}^P \)
$\mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1}$
\[ \mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( \mathsf{NP}^A \subseteq \Sigma_{k+1}^P \) if \( A \) in \( \Sigma_k^P \)
\[ \mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

- To show \( \mathsf{NP}^A \subseteq \Sigma_{k+1}^P \) if \( A \) in \( \Sigma_k^P \)

- For \( B \in \mathsf{NP}^A \) poly-time TM \( M \) s.t. \( B = \{ x \mid \exists w \, M^A(x,w)=1 \} \)
\[ \text{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( \text{NP}^{A} \subseteq \Sigma_{k+1}^P \) if \( A \) in \( \Sigma_k^P \)

- For \( B \in \text{NP}^{A} \) poly-time TM \( M \) s.t. \( B = \{ x | \exists w \ M^A(x,w)=1 \} \)
- i.e., \( B = \{ x | \exists w \ \exists \text{ans} \ M^{<\text{ans}>}(x,w)=1 \text{ and "ans correct"} \} \)
\[ \text{NP}^\Sigma_k \subseteq \Sigma_{k+1} \]

To show \( \text{NP}^A \subseteq \Sigma_{k+1}^P \) if \( A \) in \( \Sigma_k^P \)

- For \( B \in \text{NP}^A \) poly-time TM \( M \) s.t. \( B = \{ x \mid \exists w \ M^A(x,w)=1 \} \)
- i.e., \( B = \{ x \mid \exists w \ \exists \text{ans} \ M^{\text{ans}}(x,w)=1 \text{ and “ans correct”} \} \)

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and “ans correct”}\} \) in \( \Sigma_{k+1}^P \)
\[ \text{To show } \text{NP}^\Sigma_k \subseteq \Sigma_{k+1} \]

\[ \text{To show } \text{NP}^A \subseteq \Sigma_{k+1}^P \text{ if } A \text{ in } \Sigma_k^P \]

\[ \text{For } B \in \text{NP}^A \text{ poly-time TM } M \text{ s.t. } B = \{ x | \exists w \ M^A(x,w)=1 \} \]

\[\text{i.e., } B = \{ x | \exists w \ \exists \text{ans } M^{<\text{ans}>}(x,w)=1 \text{ and "ans correct"} \} \]

\[ \text{To show } C = \{ (x,w,\text{ans}) | M^{<\text{ans}>}(x,w)=1 \text{ and "ans correct"} \} \text{ in } \Sigma_{k+1}^P \]

\[ \text{Then } B \text{ also in } \Sigma_{k+1}^P \]
$\mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1}$
$\text{NP}^{\Sigma_k} \subseteq \Sigma_{k+1}$

To show $C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"}\}$ in $\Sigma_{k+1}^P$
\[ \text{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"} \} \) in \( \Sigma_{k+1}^p \)

Suppose \( M \) makes only one query \( z=Z(x,w) \). \( \text{ans} \) is a single bit saying if \( z \) in \( A \) or not.
\[ \mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"} \} \) in \( \Sigma_{k+1}^P \)

Suppose \( M \) makes only one query \( z=Z(x,w) \). \( \text{ans} \) is a single bit saying if \( z \) in \( A \) or not

“ans correct”: \((\text{ans}=1 \land z \in A)\) or \((\text{ans}=0 \land z \notin A)\)
\[ \text{NP}^\Sigma_k \subseteq \Sigma_{k+1} \]

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"} \} \) in \( \Sigma_{k+1}^p \)

Suppose \( M \) makes only one query \( z=Z(x,w) \). \( \text{ans} \) is a single bit saying if \( z \) in \( A \) or not.

"ans correct": (\( \text{ans}=1 \land z \in A \)) or (\( \text{ans}=0 \land z \notin A \))

\( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \land [(\text{ans}=1 \land \exists u_1 \forall u_2 \ldots Q_k u_k \ F(z,u_1,\ldots)=1)] \) or (\( \text{ans}=0 \land \forall v_1 \exists v_2 \ldots Q'_k v_k \ F(z,v_1,\ldots)=0) \} \} \)
$\mathsf{NP}^{\Sigma_k} \subseteq \Sigma_{k+1}$

To show $C = \{(x,w,\text{ans}) \mid \mathcal{M}^{\text{ans}}(x,w)=1 \text{ and "ans correct"}\}$ in $\Sigma_{k+1}^P$

Suppose $\mathcal{M}$ makes only one query $z = Z(x,w)$. $\text{ans}$ is a single bit saying if $z$ in $A$ or not

“ans correct”: $(\text{ans}=1 \land z \in A)$ or $(\text{ans}=0 \land z \notin A)$

$C = \{(x,w,\text{ans}) \mid \mathcal{M}^{\text{ans}}(x,w)=1 \land [(\text{ans}=1 \land \exists u_1 \forall u_2 \ldots Q_k u_k F(z,u_1,\ldots)=1) \lor (\text{ans}=0 \land \forall v_1 \exists v_2 \ldots Q’_k v_k F(z,v_1,\ldots)=0)] \}$

$C = \{(x,w,\text{ans}) \mid \exists u_1 \forall u_2 v_1 \exists u_3 v_2 \ldots Q_k u_k Q’_k v_k \mathcal{M}^{\text{ans}}(x,w)=1 \land [(\text{ans}=1 \land F(z,u_1,\ldots)=1) \lor (\text{ans}=0 \land F(z,v_1,\ldots)=0)] \}$
\[ \text{NP}^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"} \} \) in \( \Sigma_{k+1}^P \)

Suppose \( M \) makes only one query \( z=Z(x,w) \). \( \text{ans} \) is a single bit saying if \( z \) in \( A \) or not

"ans correct": \( (\text{ans}=1 \land z \in A) \text{ or } (\text{ans}=0 \land z \notin A) \)

\[ C=\{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \land [(\text{ans}=1 \land \exists u_1 \forall u_2 \ldots Q_k u_k F(z,u_1,...)=1) \]
\[ \text{or } (\text{ans}=0 \land \forall v_1 \exists v_2 \ldots Q'_k v_k F(z,v_1,...)=0)] \} \]

\[ C=\{(x,w,\text{ans}) \mid \exists u_1 \forall u_2 v_1 \exists u_3 v_2 \ldots Q_k u_k Q'_k v_k \quad M^{\text{ans}}(x,w)=1 \land \\
\[ (\text{ans}=1 \land F(z,u_1,...)=1) \text{ or } (\text{ans}=0 \land F(z,v_1,...)=0)] \} \]
\[ NP^{\Sigma_k} \subseteq \Sigma_{k+1} \]

To show \( C = \{(x,w,\text{ans}) \mid M^{\text{ans}}(x,w)=1 \text{ and "ans correct"}\} \) in \( \Sigma_{k+1}^P \)

Suppose \( M \) makes only one query \( z=Z(x,w) \). \( \text{ans} \) is a single bit saying if \( z \) in \( A \) or not

"ans correct": \((\text{ans}=1 \land z \in A) \) or \((\text{ans}=0 \land z \notin A) \)

\[ C=\{(x,w,\text{ans})\mid M^{\text{ans}}(x,w)=1 \land [(\text{ans}=1 \land \exists u_1 \forall u_2 \ldots Q_k u_k F(z,u_1,\ldots)=1) \]

or \((\text{ans}=0 \land \forall v_1 \exists v_2 \ldots Q'_k v_k F(z,v_1,\ldots)=0)] \}

\[ C=\{(x,w,\text{ans})\mid \exists u_1 \forall u_2 v_1 \exists u_3 v_2 \ldots Q_k u_k Q'_k v_k \quad M^{\text{ans}}(x,w)=1 \land \\
[(\text{ans}=1 \land F(z,u_1,\ldots)=1) \text{ or } (\text{ans}=0 \land F(z,v_1,\ldots)=0)] \}

Changes for 2 queries: \( z=Z(x,w) \rightarrow (z^{(1)},z^{(2)}) = Z(x,w,\text{ans}), \)
\( u_i \rightarrow u_i^{(1)},u_i^{(2)}, \quad v_i \rightarrow v_i^{(1)},v_i^{(2)}, \) and use conjunction of two checks (for \( j=1 \) and \( j=2 \)) of the form \[ (\text{ans}^{(j)}=1 \land F(z^{(j)},u_1^{(j)},\ldots)=1) \text{ or } (\text{ans}^{(j)}=0 \land F(z^{(j)},v_1^{(j)},\ldots)=0) \]
Oracle Version
Oracle Version

\[ \Sigma_{k+1}^P = NP^{\Sigma_k} \text{ (with } \Sigma_0^P = P) \]
Oracle Version

\[ \Sigma_{k+1}^P = \mathsf{NP}^{\Sigma_k} \text{ (with } \Sigma_0^P = \mathsf{P}) \]

\[ \Pi_{k+1}^P = \mathsf{co-NP}^{\Pi_k} \text{ (with } \Pi_0^P = \mathsf{P}) \]
Oracle Version

\[ \Sigma_{k+1}^P = \text{NP}^{\Sigma_k} \text{ (with } \Sigma_0^P = P) \]

\[ \Pi_{k+1}^P = \text{co-NP}^{\Pi_k} \text{ (with } \Pi_0^P = P) \]

\[ \Pi_{k+1}^P = \text{co-}(\text{NP}^{\Sigma_k}) = \text{co-NP}^{\Sigma_k} = \text{co-NP}^{\Pi_k} \]
$\Delta_{k^p}$
$\Delta_k^p$

$\Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k}$
$\Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k}$

$\Delta_1^p = p$
Δ_k^p

- \Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k}
- \Delta_1^p = p
- \Delta_2^p = p^{NP}
$$\Delta_k^p$$

- $$\Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k}$$
- $$\Delta_1^p = p$$
- $$\Delta_2^p = p^{NP}$$

- Note that $$\Delta_2^p = co-\Delta_2^p$$
\[ \Delta_k^p \]

- \( \Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k} \)
- \( \Delta_1^p = \text{P} \)
- \( \Delta_2^p = \text{P}^{\text{NP}} \)
- Note that \( \Delta_2^p = \text{co-}\Delta_2^p \)
- \( \Delta_{k+1}^p \supseteq \Sigma_k^p \cup \Pi_k^p \)
\[ \Delta_k^p \]

\[ \Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k} \]

\[ \Delta_1^p = p \]

\[ \Delta_2^p = p^{NP} \]

Note that \[ \Delta_2^p = \text{co-} \Delta_2^p \]

\[ \Delta_{k+1}^p \supseteq \Sigma_k^p \cup \Pi_k^p \]

\[ \Delta_{k+1}^p \subseteq \Sigma_{k+1}^p \cap \Pi_{k+1}^p \] (why?)
\( \Delta_k^p \)

- \( \Delta_{k+1}^p = p^{\Sigma_k} = p^{\Pi_k} \)
- \( \Delta_1^p = p \)
- \( \Delta_2^p = p^{NP} \)
- **Note that** \( \Delta_2^p = \text{co-}\Delta_2^p \)
- \( \Delta_{k+1}^p \supseteq \Sigma_k^p \cup \Pi_k^p \)
- \( \Delta_{k+1}^p \subseteq \Sigma_{k+1}^p \cap \Pi_{k+1}^p \) (why?)
- \( p^{\Sigma_k} \subseteq NP^{\Sigma_k} \cap \text{coNP}^{\Sigma_k} \)
PH
PH
$\mathsf{PH}$
$\text{PH}$

Diagram of the polynomial hierarchy, showing the relationships between $\Sigma_2^P$, $\Pi_2^P$, $\Sigma_3^P$, $\Pi_3^P$, $\text{NP}$, and $\text{coNP}$.
PH

\[ \Sigma_3^P \rightarrow \Sigma_2^P \rightarrow \Sigma_1^P \rightarrow \Sigma_0^P = \text{P} \]

\[ \Pi_3^P \rightarrow \Pi_2^P \rightarrow \Pi_1^P \rightarrow \Pi_0^P = \text{P} \]

\[ \Delta_2^P \rightarrow \Delta_1^P \rightarrow \Delta_0^P = \text{P} \]

\[ \text{NP} \rightarrow \Sigma_2^P \rightarrow \Sigma_3^P \rightarrow \Pi_3^P \rightarrow \text{coNP} \]

\[ \text{coNP} \rightarrow \Sigma_2^P \rightarrow \Sigma_3^P \rightarrow \Pi_3^P \rightarrow \text{NP} \]
PH
PH
Today
Today

Today, more PH
Today

- Today, more PH
- Oracle-based definitions (in particular $\text{NP}^{\text{NP}} = \Sigma^p_2$)
Today

- Today, more PH
  - Oracle-based definitions (in particular $\text{NP}^{\text{NP}} = \Sigma_2^P$)
- Next lecture, more PH
Today

Today, more PH

Oracle-based definitions (in particular $\text{NP}^{\text{NP}} = \Sigma_2^p$)

Next lecture, more PH

Alternating TM-based definitions
Today

- Today, more PH
  - Oracle-based definitions (in particular $NP^{NP} = \Sigma_2^P$)
- Next lecture, more PH
  - Alternating TM-based definitions
  - Time–Space tradeoffs