Cryptocomplexity II

Kryptokomplexität II

Sommersemester 2024

Chapter 10: Polynomial Hierarchy

Dozent: Prof. Dr. J. Rothe

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Reminder: Length-Bounded Existential Quantifier

Theorem

$A \in \mathrm{NP}$ if and only if there exist a set $B \in \mathrm{P}$ and a polynomial p such that for each $x \in \Sigma^*$,

$$x \in A \iff (\exists w)\,[|w| \leq p(|x|) \text{ and } (x,w) \in B]. \tag{5}$$

Definition (Polynomially Length-Bounded Quantifier)

Let B be a predicate, p be a polynomial, and x be a string. Define:

$$(\exists^p y)\,[B(x,y)] \iff (\exists y)\,[|y| \leq p(|x|) \text{ and } B(x,y)];$$
$$(\forall^p y)\,[B(x,y)] \iff (\forall y)\,[|y| \leq p(|x|) \text{ implies } B(x,y)].$$

For example, $\mathrm{NP} = \exists^p \cdot \mathrm{P}$ and $\mathrm{coNP} = \forall^p \cdot \mathrm{P}$.

Reminder: The Graph Isomorphism Problem

Definition (Graph Isomorphism)

Let G and H be two undirected graphs with the same number of vertices. An *isomorphism between G and H* is an edge-preserving bijection from $V(G)$ onto $V(H)$.

Formally, letting $V(G) = \{1, 2, \ldots, n\} = V(H)$, G and H are *isomorphic* ($G \cong H$, for short) if there exists a permutation $\pi \in \mathfrak{S}_n$ such that for any two vertices $i, j \in V(G)$,

$$\{i,j\} \in E(G) \iff \{\pi(i), \pi(j)\} \in E(H). \tag{6}$$

Let $\mathrm{ISO}(G, H)$ be the set of all isomorphisms between G und H.

Reminder: The Graph Isomorphism Problem

Example (Graph Isomorphism)

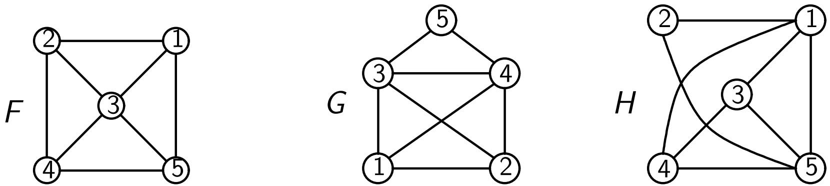


Figure: Three graphs: G is isomorphic to H, but not to F

- $G \cong H$ via $\pi = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix}$ or, in cyclic notation, by $\pi = (1\,3)(2\,4\,5)$.
- $G \not\cong F \not\cong H$: F's *sequence of vertex degrees*, $(3,3,3,3,4)$, differs from that of G and H, $(2,3,3,4,4)$.
- There are 3 more isomorphisms between G and H: $\mathrm{ISO}(G,H) = 4$.

Reminder: The Graph Isomorphism Problem

Definition (Graph Isomorphism)

The *graph isomorphism problem* (GI, for short) is defined by

$$\mathrm{GI} = \{(G,H) \mid G \text{ and } H \text{ are isomorphic graphs}\}.$$

Remark:

- The complexity status of GI is still open; it is
  - neither known to be NP-complete (though it is clearly in NP)
  - nor known to be in P.
- $\mathrm{ISO}(G,H)$ contains all *solutions* (or *witnesses*) of "$(G,H) \in \mathrm{GI}$" (with respect to the standard NPTM for solving GI):

$$\mathrm{ISO}(G,H) \neq \emptyset \iff (G,H) \in \mathrm{GI}. \tag{7}$$

Prefix Search for Finding the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine)

- **Goal**:
  - Find the lexicographically smallest isomorphism in $\mathrm{ISO}(G,H)$ if $(G,H) \in \mathrm{GI}$;
  - otherwise, "$(G,H) \notin \mathrm{GI}$" is indicated by returning the empty string ε.

- That is, we want to compute the function

$$f(G,H) = \begin{cases} \min\{\pi \mid \pi \in \mathrm{ISO}(G,H)\} & \text{if } (G,H) \in \mathrm{GI} \\ \varepsilon & \text{if } (G,H) \notin \mathrm{GI}. \end{cases}$$

Prefix Search for Finding the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- The minimum is taken w.r.t. the lexicographical order on $\mathfrak{S}_n$:
  - View a permutation $\pi \in \mathfrak{S}_n$ as the length n string $\pi(1)\pi(2)\cdots\pi(n)$ over the alphabet $[n] = \{1, 2, \ldots, n\}$.
  - For $\sigma, \tau \in \mathfrak{S}_n$, we write $\sigma < \tau$ if and only if there exists a $j \in [n]$ such that $\sigma(i) = \tau(i)$ for all $i < j$, and $\sigma(j) < \tau(j)$.
- For example, if

$$\sigma = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix} \quad \text{and} \quad \tau = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,2\,1\,5 \end{pmatrix},$$

then

$$\sigma = 34152 < 34215 = \tau.$$

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- Canceling some pairs $(i, \sigma(i))$ out of a permutation $\sigma \in \mathfrak{S}_n$, one obtains a *partial permutation*, which can also be viewed as a string over $[n] \cup \{*\}$, where $*$ indicates an undefined position.
- A *prefix of length k of $\sigma \in \mathfrak{S}_n$*, $k \leq n$, is a partial permutation of σ that contains
  - every pair $(i, \sigma(i))$ with $i \leq k$,
  - but none of the pairs $(i, \sigma(i))$ with $i > k$.
- If $k = 0$ then the empty string ε is the (unique) length 0 prefix of σ.
- If $k = n$ then the total permutation σ is the (unique) length n prefix of itself.

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- For example, if $\sigma = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix}$, then $\tau = \begin{pmatrix} 1\,3\,5 \\ 3\,1\,2 \end{pmatrix}$ is a partial permutation of σ, and $\pi = \begin{pmatrix} 1\,2\,3 \\ 3\,4\,1 \end{pmatrix}$ is a prefix of length 3 of σ.
- As a string over $[n] \cup \{*\}$, the partial permutation τ is written
$$\tau = 3 * 1 * 2.$$
For prefixes like
$$\pi = 341** = 341,$$
the placeholders $*$ may be dropped.

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- If π is a prefix of length $k < n$ of $\sigma \in \mathfrak{S}_n$ and if $w = i_1 i_2 \cdots i_{|w|}$ is a string over $[n]$ of length $|w| \leq n - k$ with none of the i_j occurring in π, then πw denotes the *partial permutation that extends π* by the pairs

$$(k+1, i_1), (k+2, i_2), \ldots, (k+|w|, i_{|w|}).$$

- If in addition $\sigma(k+j) = i_j$ for $1 \leq j \leq |w|$, then *πw is also a prefix of σ*. For example, if $\pi = \begin{pmatrix} 1\,2\,3 \\ 3\,4\,1 \end{pmatrix}$ is a prefix of $\sigma = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix}$, then
  - π is extended by each of $w_1 = 2$, $w_2 = 5$, $w_3 = 25$, and $w_4 = 52$,
  - but only $\pi w_2 = 3415$ and $\pi w_4 = 34152$ are prefixes of σ.

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- For any two graphs G and H, define the *set of prefixes of isomorphisms in* $\mathrm{ISO}(G,H)$ by

$$\text{Pre-Iso} = \{(G,H,\pi) \,\big|\, (\exists w \in [n]^*)\ [w = i_1 i_2 \cdots i_{n-|\pi|} \text{ and } \pi w \in \mathrm{ISO}(G,H)]\}.$$

- Note that
  1. for $n \geq 1$, the empty string ε does not encode a permutation in $\mathfrak{S}_n$, and
  2.

$$\begin{aligned} \mathrm{ISO}(G,H) = \emptyset \quad &\Longleftrightarrow \quad (G,H,\varepsilon) \notin \text{Pre-Iso} \\ &\Longleftrightarrow \quad (G,H) \notin \mathrm{GI} \quad \text{by (7)}. \end{aligned}$$

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

- Using $\mathrm{Pre\text{-}Iso}$ as an oracle set, the following DPOTM N computes f by prefix search.
- Thus, $f \in \mathrm{FP}^{\mathrm{Pre\text{-}Iso}}$.
- It is not difficult to prove that $\mathrm{Pre\text{-}Iso}$ is a set in NP, so $f \in \mathrm{FP}^{\mathrm{NP}}$.

Prefix Search to Find the Smallest Graph Isomorphism

Example (Prefix Search by an Oracle Turing Machine—continued)

```
N^{Pre-Iso}(G, H)  {
  if ((G, H, ε) ∉ Pre-Iso) return ε;
    else  {
        π := ε;    j := 0;
        while (j < n) {                      // G and H both have n vertices
          i := 1;
          while ((G, H, πi) ∉ Pre-Iso)  {i := i + 1;}
          π := πi;    j := j + 1;
        }
    }
  return π;
}
```

Oracle Turing Machine

Definition (Oracle Turing Machine)

- An *oracle set* (or an *oracle*, for short) is a set of strings.
- An *oracle Turing machine* (*OTM*) M is a Turing machine equipped with a special work tape (the *oracle/query tape*) and 3 special states:
  - the *query state*, $z_?$, and
  - the two *answer states* $z_{\mathtt{yes}}$ and $z_{\mathtt{no}}$.
- If not in state $z_?$, M works just like a regular Turing machine.
- However, when M reaches the query state $z_?$, it
  - interrupts its computation and
  - queries its oracle about the string q that currently is written on the oracle tape.

Oracle Turing Machine

Definition (Oracle Turing Machine—continued)

- Imagine the oracle, say B, as some kind of "black box": B answers the query of whether it contains q or not within one step of M's computation, regardless of how difficult it is to decide the set B:
  - If $q \in B$, then M changes its current state into the new state $z_{\mathtt{yes}}$, deletes the query tape, and continues its computation.
  - Otherwise (if $q \notin B$), M deletes the query tape and continues its computation in the new state $z_{\mathtt{no}}$.
- We say the computation of M on input x *is performed relative to the oracle B*, and we write $M^B(x)$.

Oracle Turing Machine

Definition (Oracle Turing Machine—continued)

- Let $L(M^B)$ be the language accepted by M^B.
- A class $\mathscr{C}$ of languages is said to be *relativizable* if it can be represented by oracle Turing machines relative to the empty oracle.
- A language $L \in \mathscr{C}$ is said to be *represented by an oracle Turing machine M* if $L = L(M^{\emptyset})$.
- For any relativizable class $\mathscr{C}$ and for any oracle B, define the class *$\mathscr{C}$ relative to B* by

$$\mathscr{C}^B = \{L(M^B) \,\big|\, M \text{ is an OTM representing some set in } \mathscr{C}\}.$$

- For any class $\mathscr{B}$ of oracle sets, define $\mathscr{C}^{\mathscr{B}} = \bigcup_{B \in \mathscr{B}} \mathscr{C}^B$.

Polynomial-Time Turing Reducibility

Definition (Turing Reducibility, Completeness, Closure)

Let $\Sigma = \{0,1\}$ be a fixed alphabet, and let A and B be sets of strings over Σ. Let $\mathscr{C}$ be any complexity class.

1. Define the *polynomial-time Turing reducibility*, denoted by $\leq_{\mathrm{T}}^{\mathrm{p}}$, as follows: $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ if and only if there is a deterministic polynomial-time oracle Turing machine (DPOTM, for short) M such that $A = L(M^B)$.
2. Define the *nondeterministic polynomial-time Turing reducibility*, denoted by $\leq_{\mathrm{T}}^{\mathrm{NP}}$, as follows: $A \leq_{\mathrm{T}}^{\mathrm{NP}} B$ if and only if there is a nondeterministic polynomial-time oracle Turing machine (NPOTM, for short) M such that $A = L(M^B)$.
3. A set B is *$\leq_{\mathrm{T}}^{\mathrm{p}}$-hard for $\mathscr{C}$* if $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ for each $A \in \mathscr{C}$.
4. A set B is *$\leq_{\mathrm{T}}^{\mathrm{p}}$-complete for $\mathscr{C}$* if B is $\leq_{\mathrm{T}}^{\mathrm{p}}$-hard for $\mathscr{C}$ and $B \in \mathscr{C}$.

Polynomial-Time Turing Reducibility

Definition (Turing Reducibility, Completeness, Closure—continued)

5. $\mathscr{C}$ is said to be *closed under the $\leq_{\mathrm{T}}^{\mathrm{p}}$-reducibility* (*$\leq_{\mathrm{T}}^{\mathrm{p}}$-closed*, for short) if for any two sets A and B, if $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ and $B \in \mathscr{C}$, then $A \in \mathscr{C}$.
6. The notions of *$\leq_{\mathrm{T}}^{\mathrm{NP}}$-hardness for $\mathscr{C}$*, *$\leq_{\mathrm{T}}^{\mathrm{NP}}$-completeness for $\mathscr{C}$*, and *$\mathscr{C}$ being $\leq_{\mathrm{T}}^{\mathrm{NP}}$-closed* are defined analogously.
7. The *Turing closure of $\mathscr{C}$* and the *$\leq_{\mathrm{T}}^{\mathrm{NP}}$-closure of $\mathscr{C}$*, respectively, are defined by:

$$\begin{aligned} \mathrm{P}^{\mathscr{C}} &= \{A \mid (\exists B \in \mathscr{C})\,[A \leq_{\mathrm{T}}^{\mathrm{p}} B]\}; \\ \mathrm{NP}^{\mathscr{C}} &= \{A \mid (\exists B \in \mathscr{C})\,[A \leq_{\mathrm{T}}^{\mathrm{NP}} B]\}. \end{aligned}$$

Properties of the Polynomial-Time Turing Reducibility

Theorem

1. $\leq_{\mathrm{m}}^{\log} \subseteq \leq_{\mathrm{m}}^{\mathrm{p}} \subseteq \leq_{\mathrm{T}}^{\mathrm{p}} \subseteq \leq_{\mathrm{T}}^{\mathrm{NP}}$.
2. *The relation* $\leq_{\mathrm{T}}^{\mathrm{p}}$ *is reflexive and transitive, yet not antisymmetric. The relation* $\leq_{\mathrm{T}}^{\mathrm{NP}}$ *is reflexive, yet neither transitive nor antisymmetric.*
3. P *and* PSPACE *are* $\leq_{\mathrm{T}}^{\mathrm{p}}$*-closed, i.e.,* $\mathrm{P}^{\mathrm{P}} = \mathrm{P}$ *and* $\mathrm{P}^{\mathrm{PSPACE}} = \mathrm{PSPACE}$.
4. $\mathrm{NP}^{\mathrm{P}} = \mathrm{NP}$ *and* $\mathrm{NP}^{\mathrm{PSPACE}} = \mathrm{PSPACE}$.
5. *If* $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ *and* A *is* $\leq_{\mathrm{T}}^{\mathrm{p}}$*-hard for a complexity class* $\mathscr{C}$*, then* B *is* $\leq_{\mathrm{T}}^{\mathrm{p}}$*-hard for* $\mathscr{C}$.
6. *If* $\mathrm{L} \neq \mathrm{NP}$*, then there exist sets* A *and* B *in* NP *such that* $A \leq_{\mathrm{T}}^{\mathrm{NP}} B$*, yet* $A \not\leq_{\mathrm{m}}^{\log} B$.

without proof

Polynomial Hierarchy

Definition (Polynomial Hierarchy)

The *polynomial hierarchy* is inductively defined by:

$$\begin{aligned} \Delta_0^p &= \Sigma_0^p = \Pi_0^p = \mathrm{P}; \\ \Delta_{i+1}^p &= \mathrm{P}^{\Sigma_i^p}, \quad \Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p}, \text{ and } \quad \Pi_{i+1}^p = \mathrm{co}\Sigma_{i+1}^p \quad \text{for } i \geq 0; \\ \mathrm{PH} &= \bigcup_{k \geq 0} \Sigma_k^p. \end{aligned}$$

Remark: In particular,

$$\begin{aligned} \Delta_1^p &= \mathrm{P}^{\Sigma_0^p} = \mathrm{P}^{\mathrm{P}} = \mathrm{P}; \\ \Sigma_1^p &= \mathrm{NP}^{\Sigma_0^p} = \mathrm{NP}^{\mathrm{P}} = \mathrm{NP}; \\ \Pi_1^p &= \mathrm{co}\Sigma_1^p = \mathrm{coNP}. \end{aligned}$$

Polynomial Hierarchy

Theorem (Meyer and Stockmeyer (1972))

1. *For each* $i \geq 0$, $\Sigma_i^p \cup \Pi_i^p \subseteq \Delta_{i+1}^p \subseteq \Sigma_{i+1}^p \cap \Pi_{i+1}^p$.
2. $\mathrm{PH} \subseteq \mathrm{PSPACE}$.
3. *Each of the classes* Δ_i^p, Σ_i^p, Π_i^p, *and* PH *is* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-closed. The* Δ_i^p *levels of the polynomial hierarchy are even closed under* $\leq_{\mathrm{T}}^{\mathrm{p}}$*-reductions.*

Proof:

1. For each class $\mathscr{C}$, we have $\mathscr{C} \subseteq \mathrm{P}^{\mathscr{C}}$, since $\leq_{\mathrm{T}}^{\mathrm{p}}$ is reflexive: If A is in $\mathscr{C}$, then $A = L(M^A)$ for some DPOTM M, so A is in $\mathrm{P}^{\mathscr{C}}$.

   Hence, $\Sigma_i^p \subseteq \mathrm{P}^{\Sigma_i^p} = \Delta_{i+1}^p$.

Polynomial Hierarchy

Since $\Delta_{i+1}^p = \mathrm{co}\Delta_{i+1}^p$, we have $\Pi_i^p = \mathrm{co}\Sigma_i^p \subseteq \Delta_{i+1}^p$. Moreover, $\Delta_{i+1}^p = \mathrm{P}^{\Sigma_i^p} \subseteq \mathrm{NP}^{\Sigma_i^p} = \Sigma_{i+1}^p$ and $\Delta_{i+1}^p = \mathrm{co}\Delta_{i+1}^p \subseteq \mathrm{co}\Sigma_{i+1}^p = \Pi_{i+1}^p$.

2. We prove by induction on i:

$$(\forall i \geq 0)\,[\Sigma_i^p \subseteq \mathrm{PSPACE}]. \tag{8}$$

The induction base, $i = 0$, is trivial: $\Sigma_0^p = \mathrm{P} \subseteq \mathrm{PSPACE}$.

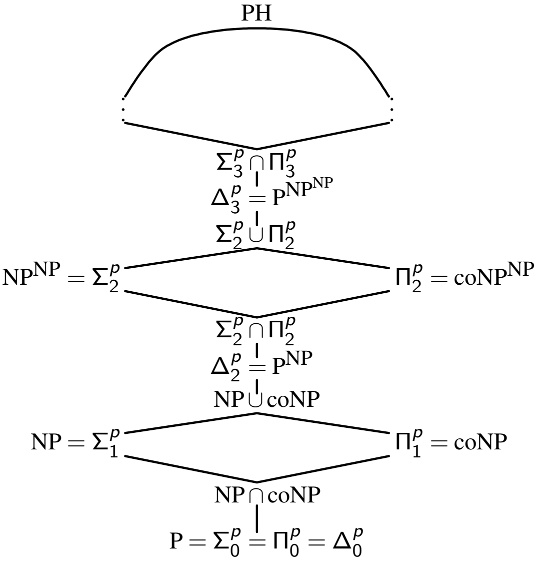
The induction hypothesis says that (8) is true for some $i \geq 0$: $\Sigma_i^p \subseteq \mathrm{PSPACE}$. Then,

$$\Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p} \subseteq \mathrm{NP}^{\mathrm{PSPACE}} \subseteq \mathrm{PSPACE},$$

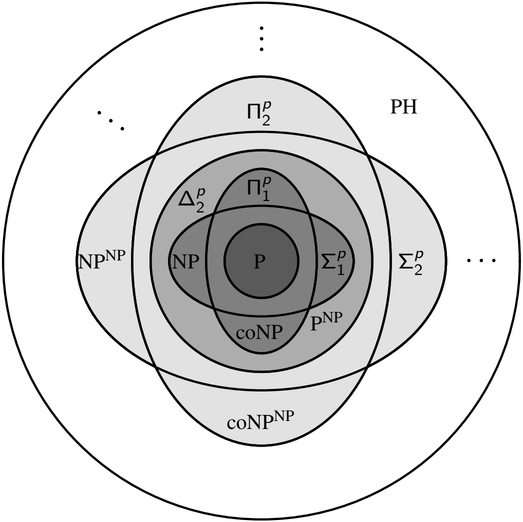
where the last inclusion can be proven analogously to the inclusion $\mathrm{NP} \subseteq \mathrm{PSPACE}$ plus a direct PSPACE simulation of the oracle queries.

3. Straightforward (left as an exercise). ❑

Polynomial Hierarchy (Hasse Diagram)



Polynomial Hierarchy (Venn Diagram)



Characterizing the Levels of the Polynomial Hierarchy

Theorem (Meyer and Stockmeyer (1972))

1. *For each* $i \geq 0$, $A \in \Sigma_i^p$ *if and only if there exist a set* $B \in \mathrm{P}$ *and a polynomial* p *such that for each* $x \in \Sigma^*$,

$$x \in A \iff (\exists^p w_1)(\forall^p w_2) \cdots (\mathfrak{Q}^p w_i)[(x, w_1, w_2, \ldots, w_i) \in B],$$

where $\mathfrak{Q}^p = \exists^p$ *if* i *is odd, and* $\mathfrak{Q}^p = \forall^p$ *if* i *is even.*

2. *For each* $i \geq 0$, $A \in \Pi_i^p$ *if and only if there exist a set* $B \in \mathrm{P}$ *and a polynomial* p *such that for each* $x \in \Sigma^*$,

$$x \in A \iff (\forall^p w_1)(\exists^p w_2) \cdots (\mathfrak{Q}^p w_i)[(x, w_1, w_2, \ldots, w_i) \in B],$$

where $\mathfrak{Q}^p = \forall^p$ *if* i *is odd, and* $\mathfrak{Q}^p = \exists^p$ *if* i *is even.*

Characterizing the Levels of the Polynomial Hierarchy

Proof: The first statement of the theorem is proven by induction on i.

The induction base $i=0$ is trivial (and case $i=1$ stated on the first slide).

The induction hypothesis says that the assertion of the theorem is true for some $i \geq 0$. We have to show that this assertion also holds for $i+1$.

$(\Rightarrow)$: Suppose that A is a set in $\Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p}$.

Let M be some NPOTM accepting A in time $q \in \mathbb{P}\mathrm{ol}$, and let $C \in \Sigma_i^p$ be M's oracle set, i.e., $A = L(M^C)$.

Define a set D as follows:

$$D = \left\{ (x,u,v,w) \;\middle|\; \begin{array}{l} w \in \mathrm{Wit}_{M^{(\cdot)}}(x),\ u = (u_1, u_2, \ldots, u_k),\ v = (v_1, v_2, \ldots, v_\ell), \\ \text{where } u \text{ gives the queries on path } w \text{ with answer ``yes''} \\ \text{and } v \text{ gives the queries on path } w \text{ with answer ``no''} \end{array} \right\}.$$

Characterizing the Levels of the Polynomial Hierarchy

Note that $D \in \mathrm{P}$. It follows from the definition of D that:

$$\begin{aligned} x \in A \quad &\Longleftrightarrow \quad M^C \text{ accepts } x \qquad (9) \\ &\Longleftrightarrow \quad (\exists^q w)\,[w \in \mathrm{Wit}_{M^C}(x)] \\ &\Longleftrightarrow \quad (\exists^q w)\,(\exists^q u)\,(\exists^q v)\,[u = (u_1, u_2, \ldots, u_k) \;\wedge\; v = (v_1, v_2, \ldots, v_\ell) \\ &\qquad \wedge\; (x,u,v,w) \in D \;\wedge\; u_1, u_2, \ldots, u_k \in C \;\wedge\; v_1, v_2, \ldots, v_\ell \notin C\,]. \end{aligned}$$

Define the sets

$$\begin{aligned} C_{\mathrm{yes}} &= \{u \,|\, u = (u_1, u_2, \ldots, u_k) \text{ and } u_1, u_2, \ldots, u_k \in C\}; \\ C_{\mathrm{no}} &= \{v \,|\, v = (v_1, v_2, \ldots, v_\ell) \text{ and } v_1, v_2, \ldots, v_\ell \notin C\}. \end{aligned}$$

Since $C \in \Sigma_i^p$, $k \leq q(|x|)$, and Σ_i^p is closed under pairing, we have $C_{\mathrm{yes}} \in \Sigma_i^p$.

Characterizing the Levels of the Polynomial Hierarchy

Similarly, since $\overline{C} \in \Pi_i^p$, $\ell \leq q(|x|)$, and Π_i^p is closed under pairing, we have $C_{\mathrm{no}} \in \Pi_i^p$.

By the induction hypothesis, for $C_{\mathrm{yes}} \in \Sigma_i^p$ and $C_{\mathrm{no}} \in \Pi_i^p$, there exist sets E and F in P and polynomials r and s such that:

$$u \in C_{\mathrm{yes}} \iff (\exists^r y_1)(\forall^r y_2) \cdots (\mathfrak{Q}^r y_i)[(u, y_1, y_2, \ldots, y_i) \in E]; \quad (10)$$

$$v \in C_{\mathrm{no}} \iff (\forall^s z_1)(\exists^s z_2) \cdots (\overline{\mathfrak{Q}}^s z_i)[(v, z_1, z_2, \ldots, z_i) \in F], \quad (11)$$

where $\mathfrak{Q}^r = \exists^r$ and $\overline{\mathfrak{Q}}^s = \forall^s$ if i is odd, and $\mathfrak{Q}^r = \forall^r$ and $\overline{\mathfrak{Q}}^s = \exists^s$ if i is even. Substituting the equivalences (10) and (11) in (9) above gives:

$$\begin{aligned} x \in A \iff & (\exists^q w)(\exists^q u)(\exists^q v)[(x, u, v, w) \in D \wedge \quad (12) \\ & (\exists^r y_1)(\forall^r y_2) \cdots (\mathfrak{Q}^r y_i)[(u, y_1, y_2, \ldots, y_i) \in E] \wedge \\ & (\forall^s z_1)(\exists^s z_2) \cdots (\overline{\mathfrak{Q}}^s z_i)[(v, z_1, z_2, \ldots, z_i) \in F]]. \end{aligned}$$

Characterizing the Levels of the Polynomial Hierarchy

Alternatingly extracting the quantifiers from the last two lines of equivalence (12) and combining contiguous equal quantifiers to one quantifier of the same type, we obtain:

$$
\begin{aligned}
& x \in A \\
\iff \quad & \underbrace{(\exists^q w)(\exists^q u)(\exists^q v)(\exists^r y_1)}_{\text{combine to } (\exists^p w_1)} \; \underbrace{(\forall^r y_2)(\forall^s z_1)}_{\text{combine to } (\forall^p w_2)} \; \cdots \underbrace{(\mathfrak{Q}^r y_i)(\mathfrak{Q}^s z_{i-1})}_{\text{combine to } (\mathfrak{Q}^p w_i)}(\overline{\mathfrak{Q}}^s z_i) \\
& [(x,u,v,w) \in D \wedge (u,y_1,y_2,\ldots,y_i) \in E \wedge (v,z_1,z_2,\ldots,z_i) \in F] \\
\iff \quad & (\exists^p w_1)(\forall^p w_2) \cdots (\overline{\mathfrak{Q}}^p w_{i+1})[(x,w_1,w_2,\ldots,w_{i+1}) \in B], \qquad (13)
\end{aligned}
$$

where $p = \max\{3q+r, r+s\}+c$ is a polynomial depending on the polynomials q, r, and s, and on a constant c, which is due to the pairing of strings when combining quantifiers.

Characterizing the Levels of the Polynomial Hierarchy

According to the quantifier combination, the set B is suitably defined so as to satisfy:

$$(x, w_1, w_2, \ldots, w_{i+1}) \in B$$
$$\iff \quad (x, u, v, w) \in D \;\wedge\; (u, y_1, y_2, \ldots, y_i) \in E \;\wedge\; (v, z_1, z_2, \ldots, z_i) \in F.$$

Since the sets D, E, and F each are in P, so is B.

By equivalence (13), A satisfies the representation (9) for $i+1$. The induction proof is complete for the direction from left to right.

Characterizing the Levels of the Polynomial Hierarchy

$(\Leftarrow)$: Suppose that there exist a set $B \in \mathrm{P}$ and a polynomial p such that A can be represented as follows:

$$A = \{x \,\big|\, (\exists^p w_1)\,(\forall^p w_2) \cdots (\mathfrak{Q}^p w_{i+1})\,[(x, w_1, w_2, \ldots, w_{i+1}) \in B],$$

where $\mathfrak{Q}^p = \exists^p$ if i is even, and $\mathfrak{Q}^p = \forall^p$ if i is odd. Define a set C by:

$$C = \{(x, w_1) \,\big|\, |w_1| \leq p(|x|) \;\wedge\; (\forall^p w_2) \cdots (\mathfrak{Q}^p w_{i+1})\,[(x, w_1, w_2, \ldots, w_{i+1}) \in B].$$

Hence,

$$x \in A \iff (\exists^p w_1)\,[(x, w_1) \in C].$$

By induction hypothesis, C is in Π_i^p; so its complement, $\overline{C}$, is in Σ_i^p.

Characterizing the Levels of the Polynomial Hierarchy

Let M be an NPOTM that, using $\overline{C}$ as an oracle, accepts A as follows:

On input x,

- nondeterministically guess a string w_1 with $|w_1| \leq p(|x|)$,
- for each w_1 guessed, query the oracle about the pair (x, w_1), and
- accept the input x if and only if the answer is "no."

It follows that $A = L(M^{\overline{C}})$. Thus, $A \in \mathrm{NP}^{\Sigma_i^p} = \Sigma_{i+1}^p$, which completes the induction proof.

The second statement of the theorem follows directly from its first statement. ❏

Polynomial Hierarchy: Upward Collapse

Theorem (Meyer and Stockmeyer (1972))

1. *For each $i \geq 0$, if $\Sigma_i^p = \Sigma_{i+1}^p$, then*
$$\Sigma_i^p = \Pi_i^p = \Delta_{i+1}^p = \Sigma_{i+1}^p = \Pi_{i+1}^p = \cdots = \mathrm{PH}.$$

2. *For each $i \geq 1$, if $\Sigma_i^p = \Pi_i^p$, then*
$$\Sigma_i^p = \Pi_i^p = \Delta_{i+1}^p = \Sigma_{i+1}^p = \Pi_{i+1}^p = \cdots = \mathrm{PH}.$$

Proof: First, we show that the hypothesis of the first statement implies that of the second statement. Supposing $\Sigma_i^p = \Sigma_{i+1}^p$ for $i \geq 0$, we have $\Pi_i^p \subseteq \Sigma_{i+1}^p = \Sigma_i^p$, which implies $\Sigma_i^p = \Pi_i^p$.

Polynomial Hierarchy: Upward Collapse

Now suppose that $\Sigma_i^p = \Pi_i^p$ for $i \geq 1$.

We show that this implies $\Sigma_i^p = \Sigma_{i+1}^p$. Let A be any set in Σ_{i+1}^p.

By the quantifier characterization theorem, there exist a set $B \in \mathrm{P}$ and a polynomial p such that for each $x \in \Sigma^*$,

$$x \in A \iff (\exists^p w_1)(\forall^p w_2) \cdots (\mathfrak{Q}^p w_{i+1})[(x, w_1, w_2, \ldots, w_{i+1}) \in B],$$

where $\mathfrak{Q}^p = \exists^p$ if i is even, and $\mathfrak{Q}^p = \forall^p$ if i is odd. Define a set C by:

$$C = \left\{ (x, w_1) \;\middle|\; \begin{array}{l} |w_1| \leq p(|x|) \;\wedge\; (\forall^p w_2)(\exists^p w_3) \cdots \\ (\mathfrak{Q}^p w_{i+1})[(x, w_1, w_2, \ldots, w_{i+1}) \in B] \end{array} \right\}.$$

Again by the quantifier characterization theorem, $C \in \Pi_i^p = \Sigma_i^p$.

Polynomial Hierarchy: Upward Collapse

Once more by the quantifier characterization theorem, for $C \in \Sigma_i^p$, there exist a set $D \in \mathrm{P}$ and a polynomial q such that for each $x \in \Sigma^*$,

$$C = \left\{ (x, w_1) \;\middle|\; \begin{array}{l} |w_1| \leq q(|x|) \;\wedge\; (\exists^q w_2)(\forall^q w_3) \cdots \\ (\overline{\mathfrak{Q}}^q w_{i+1})\,[(x, w_1, w_2, \ldots, w_{i+1}) \in D] \end{array} \right\},$$

where $\overline{\mathfrak{Q}}^q = \forall^q$ if i is even, and $\overline{\mathfrak{Q}}^q = \exists^q$ if i is odd.

Hence,

$$x \in A \iff \underbrace{(\exists^p w_1)(\exists^q w_2)}_{\text{combine to } (\exists^r w)}(\forall^q w_3) \cdots (\overline{\mathfrak{Q}}^q w_{i+1})\,[(x, w_1, w_2, \ldots, w_{i+1}) \in D].$$

Polynomial Hierarchy: Upward Collapse

Combining the first two existential quantifiers to one existential quantifier whose length is bounded by the polynomial $r = p + q$ (neglecting the constant overhead for the pairing) and once more applying the quantifier characterization theorem, we obtain $A \in \Sigma_i^p$.

Since A was arbitrarily chosen from Σ_{i+1}^p, we have $\Sigma_i^p = \Sigma_{i+1}^p$.

An easy induction now shows that every level Σ_k^p with $k \geq i$ collapses down to Σ_i^p:

$$\Sigma_{i+2}^p = \mathrm{NP}^{\Sigma_{i+1}^p} = \mathrm{NP}^{\Sigma_i^p} = \Sigma_{i+1}^p = \Sigma_i^p,$$

and so on. ❑

Quantified Boolean Formulas

Definition (Quantified Boolean Formulas)

1. Extending the set of boolean formulas, the set of *quantified boolean formulas* (*QBFs*, for short) is defined as the closure of the set of boolean constants, 0 and 1, and boolean variables, $x_1, x_2, \ldots$, under the following boolean operations:
   - $\neg$ (*negation*), $\vee$ (*disjunction*), and $\wedge$ (*conjunction*);
   - $\exists x_i$ (*existential quantification*) and $\forall x_i$ (*universal quantification*).

   Occasionally, we write $\bigvee$ for $\exists$, and $\bigwedge$ for $\forall$.

2. An occurrence of a variable x in a QBF F is said to be
   - *bound* (or *quantified*) if x occurs in a subformula of F that is of the form $(\exists x)\, G$ or $(\forall x)\, G$;
   - otherwise, this occurrence of x is *free*.

Quantified Boolean Formulas

Definition (Quantified Boolean Formulas—continued)

3. A QBF F is said to be *closed* if all variables occurring in F are quantified. Otherwise (i.e., if there occur free variables in F), F is said to be *open*.

4. The semantics of QBFs is defined in the obvious way: The notions of
   - *satisfiability*,
   - *validity*, and
   - *semantic equivalence*

   introduced for boolean formulas straightforwardly extend to quantified boolean formulas.

Reminder: Equivalences of Boolean Formulas

Table: Commonly Used Equivalences I

$\varphi_1 \vee \varphi_2$	$\equiv$	$\varphi_2 \vee \varphi_1$	Commutativity
$\varphi_1 \wedge \varphi_2$	$\equiv$	$\varphi_2 \wedge \varphi_1$	
$\neg\neg\varphi$	$\equiv$	φ	Double Negation
$(\varphi_1 \vee \varphi_2) \vee \varphi_3$	$\equiv$	$\varphi_1 \vee (\varphi_2 \vee \varphi_3)$	Associativity
$(\varphi_1 \wedge \varphi_2) \wedge \varphi_3$	$\equiv$	$\varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$	
$(\varphi_1 \wedge \varphi_2) \vee \varphi_3$	$\equiv$	$(\varphi_1 \vee \varphi_3) \wedge (\varphi_2 \vee \varphi_3)$	Distributivity
$(\varphi_1 \vee \varphi_2) \wedge \varphi_3$	$\equiv$	$(\varphi_1 \wedge \varphi_3) \vee (\varphi_2 \wedge \varphi_3)$	
$\varphi_1 \vee (\varphi_1 \wedge \varphi_2)$	$\equiv$	φ_1	Absorption Rules
$\varphi_1 \wedge (\varphi_1 \vee \varphi_2)$	$\equiv$	φ_1	

Reminder: Equivalences of Boolean Formulas

Table: Commonly Used Equivalences II

$\varphi \vee \varphi \equiv \varphi$ $\varphi \wedge \varphi \equiv \varphi$	Idempotence
$1 \vee \varphi \equiv 1$ $1 \wedge \varphi \equiv \varphi$	Tautology Rules
$0 \vee \varphi \equiv \varphi$ $0 \wedge \varphi \equiv 0$	Unsatisfiability Rules
$\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$ $\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$	De Morgan's Rules

Quantified Boolean Formulas

Remark:

- Every closed QBF F evaluates to either true or false.

- An open QBF F is a boolean function of its $k \geq 1$ free (i.e., not quantified) variables, which maps from $\{0,1\}^k$ to $\{0,1\}$.

- The equivalences for boolean formulas stated in these tables can as well be proven for quantified boolean formulas.

- Due to the addition of quantifiers in QBFs, additional equivalences can be shown. In particular, deMorgan's rule can be generalized to:

$$\neg(\exists x)\,[F(x)] \equiv (\forall x)\,[\neg F(x)] \qquad \text{and} \qquad \neg(\forall x)\,[F(x)] \equiv (\exists x)\,[\neg F(x)].$$

Quantified Boolean Formulas

Example (Closed Quantified Boolean Formulas)

Consider the closed QBF

$$G \quad = \quad (\forall x)\,[x \wedge (\exists y)\,[(x \wedge y) \Rightarrow \neg x]].$$

To evaluate G, consider the subformula

$$H(x) = (\exists y)\,[(x \wedge y) \Rightarrow \neg x]$$

of G first. The variable y is existentially quantified; assigning the truth value 0 to y thus simplifies $H(x)$ to

$$H(x) \equiv ((x \wedge 0) \Rightarrow \neg x) \equiv (0 \Rightarrow \neg x) \equiv 1.$$

Hence, G evaluates to false:

$$G \equiv (\forall x)\,[x \wedge H(x)] \equiv (\forall x)\,[x \wedge 1] \equiv (\forall x)\,[x] \equiv 0 \wedge 1 \equiv 0.$$

Quantified Boolean Formulas: Prenex Form

Definition (Prenex Form of a QBF)

A QBF F is said to be in *prenex form* if F is of the form:

$$F(x_1,\ldots,x_k) \;=\; (\mathfrak{Q}_1 y_1)\cdots(\mathfrak{Q}_n y_n)\,\varphi(x_1,\ldots,x_k,y_1,\ldots,y_n),$$

where $\mathfrak{Q}_i \in \{\exists,\forall\}$ for each i with $1 \leq i \leq n$, φ is a boolean formula without quantifiers, and $x_1,\ldots,x_k$ are the free variables occurring in F.

Quantified Boolean Formulas: Prenex Form

Example (Open Quantified Boolean Formulas)

The following open QBF is not in prenex form:

$$F(y,z) \quad = \quad (\forall x)\,(\exists y)\,[(x \wedge y) \vee \neg z] \vee \neg(\forall x)\,[x \vee \neg y]. \tag{14}$$

- The free variables of F are z and the rightmost occurrence of y;
- all other variable occurrences are quantified.

One and the same variable can occur both free and quantified in a formula.

Goal: Transform QBF $F(y,z)$ into an equivalent QBF in prenex form.

Quantified Boolean Formulas: Prenex Form

Example (Prenex Form of a QBF)

$$F(y,z) = (\forall x)\,(\exists y)\,[(x \wedge y) \vee \neg z] \vee \neg(\forall x)\,[x \vee \neg y].$$

Step 1: Rename the quantified variables to transform F into an equivalent formula F_1 in which no variable occurs both free and quantified and in which all quantified variables are disjoint:

$$F_1(y,z) = (\forall x)\,(\exists u)\,[(x \wedge u) \vee \neg z] \vee \neg(\forall v)\,[v \vee \neg y].$$

Step 2: Transform F_1 into an equivalent formula F_2 in prenex form:

$$F_2(y,z) = (\forall x)\,(\exists u)\,(\exists v)\,[(x \wedge u) \vee \neg z \vee (\neg v \wedge y)].$$

Step 3: Combine contiguous equal quantifiers in F_2 to one quantifier of the same type, which thus possibly quantifies a set of variables:

$$F_3(y,z) = (\forall x)\,(\exists\{u,v\})\,[(x \wedge u) \vee \neg z \vee (\neg v \wedge y)].$$

Quantified Boolean Formulas: Prenex Form

Example (Prenex Form of a QBF—continued)

- Note that F_3 and F are equivalent QBFs.
- To see that F_3 (and thus F) is satisfiable, choose the assignment that makes the free variables y and z true.
- Evaluating F_3 under this assignment then yields a closed QBF that can be simplified to
$$(\forall x)\,(\exists\{u, v\})\,[(x \wedge u) \vee \neg v]$$
by applying the tautology rule and the unsatisfiability rule.
- Since for each truth assignment to x, there exist truth assignments to u and v such that the subformula
$$(x \wedge u) \vee \neg v$$
evaluates to true, F_3 (and thus F) is satisfiable.

Quantified Boolean Formula Problems: QBF

Definition (Quantified Boolean Formula Problem)

Define the *quantified boolean formula problem* by:

$$\mathrm{QBF} \;=\; \{F \,|\, F \text{ is a closed QBF that evaluates to true}\}.$$

QBF Problem with a Bounded Number of Alternations

Definition ($\Sigma_i\mathrm{SAT}$)

For each $i \geq 1$, a QBF F is said to be a $\Sigma_i\mathrm{SAT}$ *formula* if F is closed and of the form:

$$F \;=\; (\exists X_1)\,(\forall X_2)\cdots(\mathfrak{Q}X_i)\,H(X_1, X_2, \ldots, X_i),$$

where

- the X_j are pairwise disjoint variable sets,
- $\mathfrak{Q} \in \{\exists, \forall\}$ and the i quantifiers alternate between $\exists$ and $\forall$,
- and H is a boolean formula without quantifiers.

For each $i \geq 1$, define the problem

$$\Sigma_i\mathrm{SAT} \;=\; \{F \mid F \text{ is a true } \Sigma_i\mathrm{SAT} \text{ formula}\}.$$

QBF Problem with a Bounded Number of Alternations

Definition ($\Pi_i\mathrm{SAT}$)

For each $i \geq 1$, a QBF F is said to be a $\Pi_i\mathrm{SAT}$ *formula* if F is closed and of the form:

$$F = (\forall X_1)(\exists X_2)\cdots(\mathfrak{Q} X_i)\, H(X_1, X_2, \ldots, X_i),$$

where

- the X_j are pairwise disjoint variable sets,
- $\mathfrak{Q} \in \{\exists, \forall\}$ and the i quantifiers alternate between $\forall$ and $\exists$,
- and H is a boolean formula without quantifiers.

For each $i \geq 1$, define the problem

$$\Pi_i\mathrm{SAT} = \{F \mid F \text{ is a true } \Pi_i\mathrm{SAT} \text{ formula}\}.$$

Complete Problems for Σ_i^p, Π_i^p, and PSPACE

Theorem

1. QBF *is* PSPACE*-complete.*
2. *For each* $i \geq 1$, $\Sigma_i\mathrm{SAT}$ *is* Σ_i^p*-complete and* $\Pi_i\mathrm{SAT}$ *is* Π_i^p*-complete.*
3. *If there exists a complete set for* PH*, then* PH *collapses down to some finite level:*

$$\mathrm{PH} = \Sigma_i^p = \Pi_i^p$$

   for some i.

without proof

Complete Problems for Σ_i^p, Π_i^p, and PSPACE

Definition (Meyer and Stockmeyer (1972))

MINIMAL:

Given: A boolean formula φ.

Question: Is it true that there exists no shorter formula equivalent to φ?

Theorem (Meyer and Stockmeyer (1972))

MINIMAL *is contained in* $\Pi_2^p = \mathrm{coNP}^{\mathrm{NP}}$. **without proof**

Theorem (Hemaspaandra and Wechsung (2002))

MINIMAL *is* coNP*-hard.* **without proof**

Complete Problems for Σ_i^p, Π_i^p, and PSPACE

Definition (Garey and Johnson (1979) & Stockmeyer (1977))

Minimum Equivalent Expression (MEE):

Given: A boolean formula φ and a nonnegative integer k.

Question: Does there exist a boolean formula ψ with at most k literals such that ψ is equivalent to φ?

Minimum Equivalent DNF Expression (MEE-DNF):

Given: A boolean formula φ in DNF and a nonnegative integer k.

Question: Does there exist a boolean formula ψ in DNF with at most k literals such that ψ is equivalent to φ?

Complete Problems for Σ_i^p, Π_i^p, and PSPACE

Fact

MEE *and* $\mathrm{MEE\text{-}DNF}$ *are contained in* $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$. **without proof**

Theorem (Hemaspaandra and Wechsung (2002))

MEE *and* $\mathrm{MEE\text{-}DNF}$ *are* $\mathrm{P}_{\|}^{\mathrm{NP}}$*-hard, where* $\mathrm{P}_{\|}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP[log]}} = \Theta_2^p$ *denotes the restriction of* $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$ *to "parallel" oracle access.* **without proof**

Theorem (Umans (2001))

$\mathrm{MEE\text{-}DNF}$ *is* Σ_2^p*-complete.* **without proof**

Remark: The complexity of MINIMAL and MEE is still open.

$\mathrm{BH}(\mathrm{NP})$ versus PH

Definition

Let $\mathrm{P}^{\mathrm{NP}[\mathscr{O}(1)]}$ denote the restriction of $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$ to those problems that can be solved by a DPOTM asking at most a constant number of queries to the NP oracle.

Theorem

$\mathrm{BH}(\mathrm{NP}) = \mathrm{P}^{\mathrm{NP}[\mathscr{O}(1)]}$. **without proof**

Sparse Language

Definition

- For any language S and any $n \in \mathbb{N}$, define the *set of strings of length up to n* by
$$S^{\leq n} = \{x \mid x \in S \text{ and } |x| \leq n\}.$$
- A language S is said to be *sparse* if
$$(\exists p \in \mathrm{IPol})\,(\forall n \in \mathbb{N})\ \left[\|S^{\leq n}\| \leq p(n)\right].$$

Lemma (Yap (1983))

If there exists a sparse set S such that $\mathrm{coNP} \subseteq \mathrm{NP}^S$, then
$$\mathrm{PH} = \Sigma_3^p \cap \Pi_3^p.$$

without proof

The Boolean Hierarchy Collapses the Polynomial Hierarchy

Theorem (Kadin (1988))

If there is some $k \geq 1$ such that $\mathrm{BH}_k(\mathrm{NP}) = \mathrm{coBH}_k(\mathrm{NP})$, then the polynomial hierarchy collapses down to its third level:

$$\mathrm{PH} = \Sigma_3^p \cap \Pi_3^p.$$ **without proof**

Relativized Results

Theorem

There exists an oracle A such that

$$\mathrm{P}^A = \mathrm{NP}^A = \mathrm{coNP}^A = \mathrm{PSPACE}^A.$$

Theorem

There exists an oracle A such that

$$\mathrm{P}^A \neq \mathrm{NP}^A.$$ **without proof**

Relativized Results

Theorem

There exists an oracle A such that $\mathrm{NP}^A \neq \mathrm{coNP}^A$. **without proof**

Remark: Combining these oracle constructions, we get oracles A and B such that

$$\begin{array}{ccccc} \mathrm{P}^A & \neq & \mathrm{NP}^A & \neq & \mathrm{coNP}^A \\ \mathrm{P}^B & = & \mathrm{NP}^B & \neq & \mathrm{coNP}^B. \end{array}$$

Theorem (Baker, Gill, and Solovay (1975))

There exists an oracle A such that

$$\mathrm{P}^A = \mathrm{NP}^A \cap \mathrm{coNP}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A.$$

without proof

Relativized Results

Remark: Further relativized results:

- Baker & Selman (1979): There exists an oracle A such that

$$\mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{NP}^{\mathrm{NP}^A} = \Sigma_2^{p,A}.$$

- Yao (1982): There exists an oracle A such that

$$(\forall i \geq 0)\left[\Sigma_i^{p,A} \neq \Sigma_{i+1}^{p,A}\right].$$

- Random oracles: probability measure on $\mathfrak{P}(\Sigma^*)$, the set of all oracles.
  - Bennett & Gill (1981):

$$\mathrm{Prob}(\{A \,|\, \mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A\}) = 1.$$

  - Cai (1988):

$$\mathrm{Prob}(\{A \,|\, \mathrm{PH}^A \neq \mathrm{PSPACE}^A\}) = 1.$$