Polynomial hierarchy

In computational complexity theory, the **polynomial hierarchy** (sometimes called the **polynomial-time hierarchy**) is a hierarchy of complexity classes that generalize the classes P, NP and co-NP to oracle machines. It is a resource-bounded counterpart to the arithmetical hierarchy and analytical hierarchy from mathematical logic.

1 Definitions

There are multiple equivalent definitions of the classes of the polynomial hierarchy.

1. For the oracle definition of the polynomial hierarchy, define

   $\Delta_0^{\mathrm{P}} := \Sigma_0^{\mathrm{P}} := \Pi_0^{\mathrm{P}} := \mathrm{P},$

   where P is the set of decision problems solvable in polynomial time. Then for i ≥ 0 define

   $\Delta_{i+1}^{\mathrm{P}} := \mathrm{P}^{\Sigma_i^{\mathrm{P}}}$

   $\Sigma_{i+1}^{\mathrm{P}} := \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$

   $\Pi_{i+1}^{\mathrm{P}} := \mathrm{coNP}^{\Sigma_i^{\mathrm{P}}}$

   where P^{A} is the set of decision problems solvable in polynomial time by a Turing machine augmented by an oracle for some complete problem in class A; the classes NP^{A} and $\mathrm{coNP}^{\mathrm{A}}$ are defined analogously. For example, $\Sigma_1^{\mathrm{P}} = \mathrm{NP}, \Pi_1^{\mathrm{P}} = \mathrm{coNP}$, and $\Delta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}}$ is the class of problems solvable in polynomial time with an oracle for some NP-complete problem.

2. For the existential/universal definition of the polynomial hierarchy, let L be a language (i.e. a decision problem, a subset of {0,1}*), let p be a polynomial, and define

   $$\exists^p L := \left\{ x \in \{0,1\}^* \;\middle|\; \left(\exists w \in \{0,1\}^{\leq p(|x|)}\right) \langle x, w\rangle \in L \right\},$$

   where $\langle x, w\rangle \in \{0,1\}^*$ is some standard encoding of the pair of binary strings x and w as a single binary string. L represents a set of ordered pairs of strings, where the first string x is a member of $\exists^p L$, and the second string w is a "short" ($|w| \leq p(|x|)$) witness testifying that x is a member of $\exists^p L$. In other words, $x \in \exists^p L$ if and only if there exists a short witness w such that $\langle x, w\rangle \in L$. Similarly, define

   $$\forall^p L := \left\{ x \in \{0,1\}^* \;\middle|\; \left(\forall w \in \{0,1\}^{\leq p(|x|)}\right) \langle x, w\rangle \in L \right\}$$

   Note that De Morgan's laws hold: $(\exists^p L)^{\mathrm{c}} = \forall^p L^{\mathrm{c}}$ and $(\forall^p L)^{\mathrm{c}} = \exists^p L^{\mathrm{c}}$, where L^{c} is the complement of L.

   Let $\mathcal{C}$ be a class of languages. Extend these operators to work on whole classes of languages by the definition

   $\exists^{\mathrm{P}}\mathcal{C} := \{\exists^p L \mid p \text{ is a polynomial and } L \in \mathcal{C}\}$

   $\forall^{\mathrm{P}}\mathcal{C} := \{\forall^p L \mid p \text{ is a polynomial and } L \in \mathcal{C}\}$

   Again, De Morgan's laws hold: $\mathrm{co}\exists^{\mathrm{P}}\mathcal{C} = \forall^{\mathrm{P}}\mathrm{co}\mathcal{C}$ and $\mathrm{co}\forall^{\mathrm{P}}\mathcal{C} = \exists^{\mathrm{P}}\mathrm{co}\mathcal{C}$, where $\mathrm{co}\mathcal{C} = \{L^c | L \in \mathcal{C}\}$.

   The classes **NP** and **co-NP** can be defined as $\mathrm{NP} = \exists^{\mathrm{P}}\mathrm{P}$, and $\mathrm{coNP} = \forall^{\mathrm{P}}\mathrm{P}$, where **P** is the class of all feasibly (polynomial-time) decidable languages. The polynomial hierarchy can be defined recursively as

   $\Sigma_0^{\mathrm{P}} := \Pi_0^{\mathrm{P}} := \mathrm{P}$

   $\Sigma_{k+1}^{\mathrm{P}} := \exists^{\mathrm{P}}\Pi_k^{\mathrm{P}}$

   $\Pi_{k+1}^{\mathrm{P}} := \forall^{\mathrm{P}}\Sigma_k^{\mathrm{P}}$

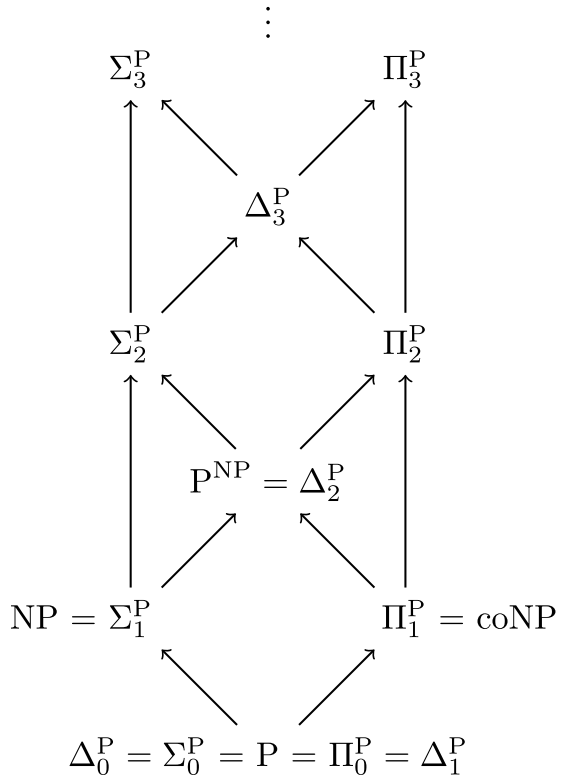
   Note that $\mathrm{NP} = \Sigma_1^{\mathrm{P}}$, and $\mathrm{coNP} = \Pi_1^{\mathrm{P}}$.

   This definition reflects the close connection between the polynomial hierarchy and the arithmetical hierarchy, where **R** and **RE** play roles analogous to **P** and **NP**, respectively. The analytic hierarchy is also defined in a similar way to give a hierarchy of subsets of the real numbers.

3. An equivalent definition in terms of alternating Turing machines defines Σ_k^{P} (respectively, Π_k^{P}) as the set of decision problems solvable in polynomial time on an alternating Turing machine with k alternations starting in an existential (respectively, universal) state.

2 Relations between classes in the polynomial hierarchy

The definitions imply the relations:



Commutative diagram equivalent to the polynomial time hierarchy. The arrows denote inclusion.

$\Sigma_i^{\mathrm{P}} \subseteq \Delta_{i+1}^{\mathrm{P}} \subseteq \Sigma_{i+1}^{\mathrm{P}}$

$\Pi_i^{\mathrm{P}} \subseteq \Delta_{i+1}^{\mathrm{P}} \subseteq \Pi_{i+1}^{\mathrm{P}}$

$\Sigma_i^{\mathrm{P}} = \mathrm{co}\Pi_i^{\mathrm{P}}$

Unlike the arithmetic and analytic hierarchies, whose inclusions are known to be proper, it is an open question whether any of these inclusions are proper, though it is widely believed that they all are. If any $\Sigma_k^{\mathrm{P}} = \Sigma_{k+1}^{\mathrm{P}}$, or if any $\Sigma_k^{\mathrm{P}} = \Pi_k^{\mathrm{P}}$, then the hierarchy *collapses to level k*: for all $i > k$, $\Sigma_i^{\mathrm{P}} = \Sigma_k^{\mathrm{P}}$. In particular, if P = NP, then the hierarchy collapses completely.

The union of all classes in the polynomial hierarchy is the complexity class **PH**.

3 Properties

The polynomial hierarchy is an analogue (at much lower complexity) of the exponential hierarchy and arithmetical hierarchy.

It is known that PH is contained within PSPACE, but it is not known whether the two classes are equal. One useful reformulation of this problem is that PH = PSPACE if and only if second-order logic over finite structures gains no additional power from the addition of a transitive closure operator.

If the polynomial hierarchy has any complete problems, then it has only finitely many distinct levels. Since there are PSPACE-complete problems, we know that if PSPACE = PH, then the polynomial hierarchy must collapse, since a PSPACE-complete problem would be a Σ_k^{P} -complete problem for some *k*.

Each class in the polynomial hierarchy contains $\leq_{\mathrm{m}}^{\mathrm{P}}$ -complete problems (problems complete under polynomial-time many-one reductions). Furthermore, each class in the polynomial hierarchy is *closed under* $\leq_{\mathrm{m}}^{\mathrm{P}}$ *-reductions*: meaning that for a class $\mathcal{C}$ in the hierarchy and a language $L \in \mathcal{C}$, if $A \leq_{\mathrm{m}}^{\mathrm{P}} L$, then $A \in \mathcal{C}$ as well. These two facts together imply that if K_i is a complete problem for Σ_i^{P} , then $\Sigma_{i+1}^{\mathrm{P}} = \mathrm{NP}^{K_i}$, and $\Pi_{i+1}^{\mathrm{P}} = \mathrm{coNP}^{K_i}$. For instance, $\Sigma_2^{\mathrm{P}} = \mathrm{NP}^{\mathrm{SAT}}$. In other words, if a language is defined based on some oracle in $\mathcal{C}$, then we can assume that it is defined based on a complete problem for $\mathcal{C}$. Complete problems therefore act as "representatives" of the class for which they are complete.

The Sipser–Lautemann theorem states that the class BPP is contained in the second level of the polynomial hierarchy.

Kannan's theorem states that for any *k*, Σ_2 is not contained in **SIZE**(n^k).

Toda's theorem states that the polynomial hierarchy is contained in $\mathrm{P}^{\#\mathrm{P}}$.

4 Problems in the polynomial hierarchy

- An example of a natural problem in Σ_2^{P} is *circuit minimization*: given a number *k* and a circuit *A* computing a Boolean function *f*, determine if there is a circuit with at most *k* gates that computes the same function *f*. Let $\mathcal{C}$ be the set of all boolean circuits. The language

$L = \{\langle A, k, B, x\rangle \in \mathcal{C} \times \mathbb{N} \times \mathcal{C} \times \{0,1\}^* \mid B \text{ has at most } k \text{ gates, and } A($

is decidable in polynomial time. The language

$$CM = \left\{ \langle A, k\rangle \in \mathcal{C} \times \mathbb{N} \;\middle|\; \begin{array}{l} \text{there exists a circuit } B \text{ with at most } k \text{ gates} \\ \text{such that } A \text{ and } B \text{ compute the same function} \end{array} \right\}$$

is the circuit minimization language. $CM \in \Sigma_2^P (= \exists^{\mathrm{P}}\forall^{\mathrm{P}}\mathrm{P})$ because L is decidable in polynomial time and because, given $\langle A, k\rangle$, $\langle A, k\rangle \in CM$ if and only if *there exists* a circuit B such that *for all* inputs x , $\langle A, k, B, x\rangle \in L$.

- A complete problem for Σ_k^{P} is **satisfiability for quantified Boolean formulas with *k* alternations of quantifiers** (abbreviated $\mathbf{QBF_k}$ or $\mathbf{QSAT_k}$).

This is the version of the boolean satisfiability problem for Σ_k^{P} . In this problem, we are given a Boolean formula f with variables partitioned into k sets X_1, ..., Xk. We have to determine if it is true that

$\exists X_1 \forall X_2 \exists X_3 \ldots f$

That is, is there an assignment of values to variables in X_1 such that, for all assignments of values in X_2, there exists an assignment of values to variables in X_3, ... f is true?

The variant above is complete for Σ_k^{P} . The variant in which the first quantifier is "for all", the second is "exists", etc., is complete for Π_k^{P} .

5 See also

- EXPTIME
- Exponential hierarchy
- Arithmetic hierarchy

6 References

1. A. R. Meyer and L. J. Stockmeyer. The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space. *In Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pp. 125–129, 1972. The paper that introduced the polynomial hierarchy.
2. L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, vol.3, pp. 1–22, 1976.
3. C. Papadimitriou. Computational Complexity. Addison-Wesley, 1994. Chapter 17. *Polynomial hierarchy*, pp. 409–438.
4. Michael R. Garey and David S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman. ISBN 0-7167-1045-5. Section 7.2: The Polynomial Hierarchy, pp. 161–167.

7 Text and image sources, contributors, and licenses

7.1 Text

- **Polynomial hierarchy** *Source:* https://en.wikipedia.org/wiki/Polynomial_hierarchy?oldid=756460526 *Contributors:* Michael Hardy, Ehn, Dcoetzee, Altenmann, MathMartin, Andris, Gdr, Creidieki, EmilJ, Oliphaunt, Ruud Koot, Dillon256, Ian Pitchford, Florian Huber, Pexatus, DVdm, Roboto de Ajvol, Wavelength, Ott2, Little Savage, That Guy, From That Show!, SmackBot, Eskimbot, Khazar, CRGreathouse, Joshwa, Pascal.Tesson, Joachim Selke, Magioladitis, A3nm, Jamelan, SieBot, Bender2k14, Addbot, RPHv, Citation bot, Xqbot, Miym, Omnipaedista, SassoBot, RobinK, Ulbra, Ripchip Bot, Frietjes, Subshift, Ethically Yours, Deas cz, Flof, Ted.tem.parker and Anonymous: 23

7.2 Images

- **File:Polynomial_time_hierarchy.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/9e/Polynomial_time_hierarchy.svg *License:* CC BY-SA 3.0 *Contributors:* Created in LaTeX by the following code:
\documentclass{article} \thispagestyle{empty} \usepackage{amsmath} \usepackage{tikz} \usetikzlibrary{arrows} \begin{document} \begin{tikzpicture}[->, node distance=2cm, semithick] \node (P) {$\Delta_0^\text{P} =\Sigma_0^\text{P}$ = P = $\Pi_0^\text{P} = \Delta_1^\text{P}$}; \node (Sigma1) [above left of=P] {NP = Σ_1^P \hspace*{0.9cm}}; \node (Pi1) [above right of=P] {\hspace*{1.2cm} Π_1^P = coNP}; \node (Delta2) [above left of=Pi1] {$\text{P}^\text{NP} = \Delta_2^\text{P}$}; \node (Sigma2) [above left of=Delta2] {Σ_2^P}; \node (Pi2) [above right of=Delta2] {Π_2^P}; \node (Delta3) [above left of=Pi2] {Δ_3^P}; \node (Sigma3) [above left of=Delta3] {Σ_3^P}; \node (Pi3) [above right of=Delta3] {$\Pi_3^\text{P}<span class="s">$}; \node (dots) [above of=Delta3] {\vdots}; \draw (P) -> (Sigma1); \draw (P) -> (Pi1); \draw (Sigma1) -> (Sigma2); \draw (Sigma1) -> (Delta2); \draw (Pi1) -> (Pi2); \draw (Pi1) -> (Delta2); \draw (Delta2) -> (Sigma2); \draw (Delta2) -> (Pi2); \draw (Sigma2) -> (Sigma3); \draw (Sigma2) -> (Delta3); \draw (Pi2) -> (Pi3); \draw (Pi2) -> (Delta3); \draw (Delta3) -> (Sigma3); \draw (Delta3) -> (Pi3); \end{tikzpicture} \end{document}

  Original artist: Self

7.3 Content license

- Creative Commons Attribution-Share Alike 3.0