The Polynomial Hierarchy Three Ways

We discovered a hierarchy of complexity classes between P and PSpace, with NP and coNP on the first level, and infinitely many further levels above:

**Definition by ATM:** Classes \( \Sigma^p_i / \Pi^p_i \) are defined by polytime ATMs with bounded types of alternation, starting computation with existential/universal states.

**Definition by Verifier:** Classes \( \Sigma^p_i / \Pi^p_i \) are given as projections of certain verifier languages in P, requiring existence/universality of polynomial witnesses.

**Definition by Oracle:** Classes \( \Sigma^p_i / \Pi^p_i \) are defined as languages of NP/coNP oracle TMs with \( \Sigma^p_{i-1} \) (or, equivalently, \( \Pi^p_{i-1} \)) oracle.

Using such oracles with deterministic TMs, we can also define classes \( \Delta^p_i \).

Is the Polynomial Hierarchy Real?

**Questions:**

- Are all of these classes really distinct? Nobody knows.
- Are any of these classes really distinct? Nobody knows.
- Are any of these classes distinct from P? Nobody knows.
- Are any of these classes distinct from PSpace? Nobody knows.

**What do we know then?**
**What We Know (Excerpt)**

**Theorem 18.1:** If there is any $k$ such that $\Sigma^P_k = \Sigma^P_{k+1}$ then $\Sigma^P_j = \Pi^P_j = \Sigma^P_k$ for all $j > k$, and therefore $\text{PH} = \Sigma^P_k$. In this case, we say that the polynomial hierarchy collapses at level $k$.

*Proof:* Left as exercise (not too hard to get from definitions). □

**Corollary 18.2:** If $\text{PH} \neq \text{P}$ then $\text{NP} \neq \text{P}$.

Intuitively speaking: “The polynomial hierarchy is built upon the assumption that NP has some additional power over P. If this is not the case, the whole hierarchy collapses.”

**Theorem 18.3:** $\text{PH} \subseteq \text{PSpace}$.

*Proof:* Left as exercise (induction over PH levels, using that $\text{PSpace} \subseteq \text{PSpace}$). □

**Theorem 18.4:** If $\text{PH} = \text{PSpace}$ then there is some $k$ with $\text{PH} = \Sigma^P_k$.

*Proof:* If $\text{PH} = \text{PSpace}$ then $\text{True QBF} \in \text{PH}$. Hence $\text{True QBF} \in \Sigma^P_k$ for some $k$. Since $\text{True QBF}$ is PSpace-hard, this implies $\Sigma^P_k = \text{PSpace}$. □

**What We Believe (Excerpt)**

“Most experts” think that:
- The polynomial hierarchy does not collapse completely (same as $\text{P} \neq \text{NP}$)
- The polynomial hierarchy does not collapse on any level (in particular $\text{PH} \neq \text{PSpace}$ and there is no PH-complete problem)

But there can always be surprises . . .

**Question 1: The Logarithmic Hierarchy**
Q1: The Logarithmic Hierarchy

The Polynomial Hierarchy is based on polynomially time-bounded TMs

It would also be interesting to study the Logarithmic Hierarchy obtained by considering logarithmically space-bounded TMs instead, wouldn't it?

In detail, we can define:

- \( \Sigma^L_0 = \Pi^L_0 = L \)
- \( \Sigma^L_i = NL^{\Sigma^L_{i-1}} \) alternatively: languages of log-space bounded \( \Sigma^L_{i+1} \) ATMs
- \( \Pi^L_i = coNL^{\Sigma^L_{i-1}} \) alternatively: languages of log-space bounded \( \Pi^L_{i+1} \) ATMs

Therefore \( \Sigma^L_i = \Pi^L_i = NL \) for all \( i \geq 1 \).

The Logarithmic Hierarchy collapses on the first level.

Historic note: In 1987, just before the Immerman-Szelepcsényi Theorem was published, Klaus-Jörn Lange, Birgit Jenner, and Bernd Kirsig showed that the Logarithmic Hierarchy collapses on the second level [ICALP 1987].

Q2: The Hardest Problems in P

What we know about P and NP:

- We don't know if any problem in NP is really harder than any problem in P.
- But we do know that NP is at least as challenging as P, i.e., \( P \subseteq NP \).

So all problems that are hard for NP are also hard for P, aren't they?
Q2: Is NP-hard as hard as P-hard?

Let’s first recall the definitions:

**Definition:** A problem $L$ is NP-hard if, for all problems $M \in NP$, there is a polynomial many-one reduction $M \leq_T L$.

**Definition:** A problem $L$ is P-hard if, for all problems $M \in P$, there is a log-space reduction $M \leq_L L$.

How to show “NP-hard implies P-hard”?
- Assume that $L$ is NP-hard.
- Consider any language $M \in P$.
- Then $M \in NP$.
- So there is a polynomial many-one reduction $f$ from $M$ to $L$.
- Hence, . . . well . . . , nothing much, really.

Q2: Is NP-hard as hard as P-hard?

For all we know today, it is perfectly possible that there are NP-hard problems that are not P-hard.

**Example 18.5:** We know that $L \subseteq P \subseteq NP$ but we do not know if any of these subsumptions are proper. Suppose that the truth actually looks like this: $L \subset P = NP$. Then all non-trivial problems in $P$ are NP-hard (why?), but not every problem would be P-hard (why?).

**Note:** This is really about the different notions of reduction used to define hardness. If we used log-space reductions for P-hardness and NP-hardness, the claim would follow.

Q3: Problems harder than P

Polynomial time is an approximation of “practically tractable” problems:
- Many practical problems are in $P$, including many very simple ones (e.g., $\emptyset$).
- P-hard problems are as hard as any other problem in $P$, and P-complete problems therefore are the hardest problems in $P$.
- However, there are even harder problems that are no longer in $P$.

So, clearly, problems that are not even in $P$ must be P-hard, right?
Q3: Are problems harder than P also hard for P?

Can we find any problem that is surely harder than P? Yes, easily:

- The Halting Problem is undecidable and therefore not in P
- Any ExpTime-complete problem is not in P (Time Hierarchy Theorem); e.g., the Word Problem for exponentially time-bounded DTMs

These concrete examples both are hard for P:

- The Word Problem for polynomially time-bounded DTMs is hard for P
- This polytime Word Problem log-space reduces to the Word Problem for exponential TMs (reduction: the identity function)
- It also log-space reduces to the Halting problem: a reduction merely has to modify the TM so that every rejecting halting configuration leads into an infinite loop

Q4: Are decidable problems harder than P also hard for P?

Schöning to the rescue (see Theorem 15.2):

**Corollary 18.6:** Consider the classes \( C_1 = \text{ExpPHard} \) (P-hard problems in Exp-Time) and \( C_2 = \text{P} \). Both are classes of decidable languages. We find that for either class \( C_k \):

- We can effectively enumerate TMs \( M^i_k \); \( i \geq 0 \);
- If \( L \in C_k \) and \( L' \) differs from \( L \) on only a finite number of words, then \( L' \in C_k \).

Let \( L_1 = \emptyset \), and let \( L_2 \) be some ExpTime-complete problem. Clearly, \( L_1 \notin \text{ExpPHard} \) and \( L_2 \notin \text{P} \) (Time Hierarchy), hence there is a decidable language \( L_d \notin \text{ExpPHard} \cup \text{P} \).

Moreover, as \( \emptyset \in \text{P} \) and \( L_2 \) is not trivial, \( L_d \preceq_p L_2 \) and hence \( L_d \in \text{ExpTime} \).

Therefore \( L_d \notin \text{ExpPHard} \) implies that \( L_d \) is not P-hard.

No, there are problems in ExpTime that are neither in P nor hard for P.

(Other arguments can even show the existence of undecidable sets that are not P-hard)

Discussion:

- Considering Questions 3 and 4, the use of the word hard is misleading, since we interpret it as difficult
- However, the actual meaning difficult would be “not in a given class” (e.g., problems not in P are clearly more difficult than those in P)
- Our formal notion of hard also implies that a problem is difficult in some sense, but it also requires it to be universal in the sense that many other problems can be solved through it

What we have seen is that there are difficult problems that are not universal.
Summary and Outlook

We do not know if the Polynomial Hierarchy is real or collapses.

Answer 1: The Logarithmic Hierarchy collapses.

Answer 2: We don’t know that NP-hard implies P-hard.

Answer 3: Being outside of P does not make a problem P-hard.

What’s next?
- Holidays
- Circuits as an alternative model of computation
- Randomness

Here’s wishing you
a Merry Christmas, a Happy Hanukkah,
a Joyous Yalda, a Cheerful Dōngzhì,
a Great Feast of Juul,
and a Wonderful Winter Solstice,
respectively!