Computability and Complexity
COSC 4200

The Polynomial-Time Hierarchy

Unique Satisfiability Problem

Consider the following decision problem:

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

A formula ϕ is in UNIQUE-SAT if it is satisfiable and has exactly one satisfying assignment.

Unique Satisfiability Problem

Consider the following decision problem:

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

A formula ϕ is in UNIQUE-SAT if it is satisfiable and has exactly one satisfying assignment.

We can put $\text{UNIQUE-SAT} \in \text{PSPACE}$ by checking all assignments and counting how many are satisfying.

Can we improve this to NP or coNP?

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

There does not seem to be a way to express UNIQUE-SAT with only existential or only universal quantifiers.

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

There does not seem to be a way to express UNIQUE-SAT with only existential or only universal quantifiers.

However, we can express it using both:

$$\begin{aligned} &\phi \in \text{UNIQUE-SAT} \\ &\iff (\exists \tau) \left[\begin{array}{l} \tau \text{ satisfies } \phi \\ \text{and} \\ (\forall \tau')\ \tau = \tau' \text{ or } \tau' \text{ does not satisfy } \phi \end{array} \right] \end{aligned}$$

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

There does not seem to be a way to express UNIQUE-SAT with only existential or only universal quantifiers.

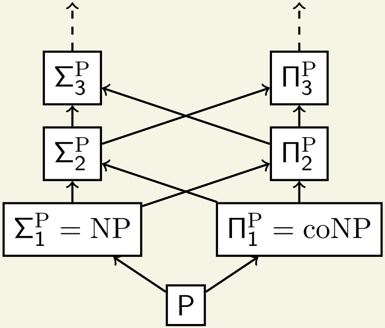
However, we can express it using both:

$$\begin{aligned} &\phi \in \text{UNIQUE-SAT} \\ &\iff (\exists \tau) \left[\begin{array}{l} \tau \text{ satisfies } \phi \\ \text{and} \\ (\forall \tau')\ \tau = \tau' \text{ or } \tau' \text{ does not satisfy } \phi \end{array} \right] \end{aligned}$$

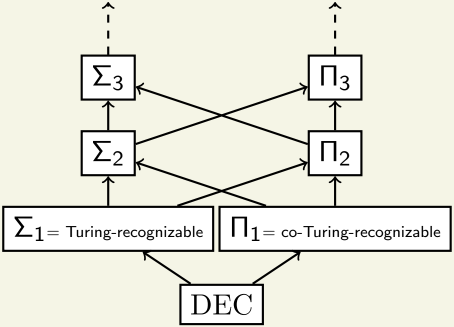
The *polynomial-time hierarchy* allows us to classify UNIQUE-SAT. The hierarchy consists of classes defined by putting combinations of $\exists$ and $\forall$ quantifiers in front of P.

The Polynomial-Time Hierarchy

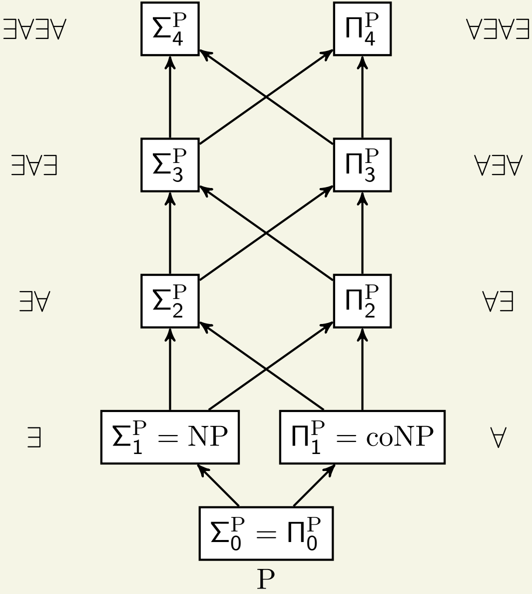
The Polynomial-Time Hierarchy is a collection of complexity classes that sit about NP and coNP. It is an analogue of the Arithmetical Hierarchy.



The Polynomial-Time Hierarchy



The Arithmetical Hierarchy



The Polynomial-Time Hierarchy

First Level

$A \in \Sigma_1^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and a polynomial p such that for all n and all $x \in \Sigma^n$,

$$x \in A \iff (\exists w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in B.$$

$\Sigma_1^{\mathrm{P}} = \mathrm{NP}$

First Level

$A \in \Sigma_1^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and a polynomial p such that for all n and all $x \in \Sigma^n$,

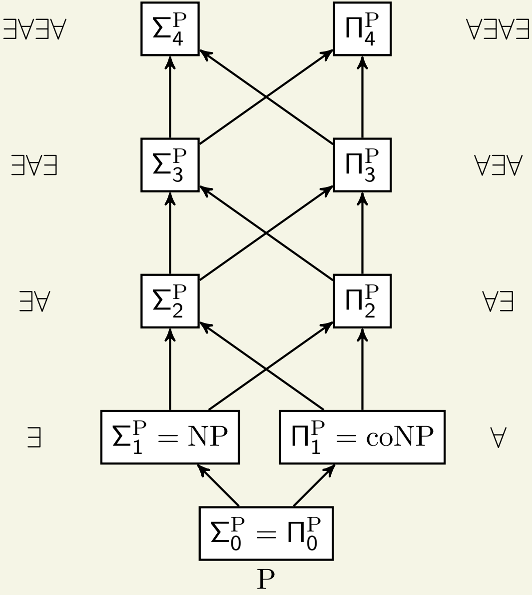
$$x \in A \iff (\exists w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in B.$$

$\Sigma_1^{\mathrm{P}} = \mathrm{NP}$

$A \in \Pi_1^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and a polynomial p such that for all n and all $x \in \Sigma^n$,

$$x \in A \iff (\forall w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in B.$$

$\Pi_1^{\mathrm{P}} = \mathrm{coNP}$



The Polynomial-Time Hierarchy

Second Level

$A \in \Sigma_2^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and polynomials p, q such that for all n and all $x \in \Sigma^n$,

$$x \in A \iff (\exists w \in \Sigma^{\leq q(n)})(\forall y \in \Sigma^{\leq p(n)})\ \langle x, w, y \rangle \in B.$$

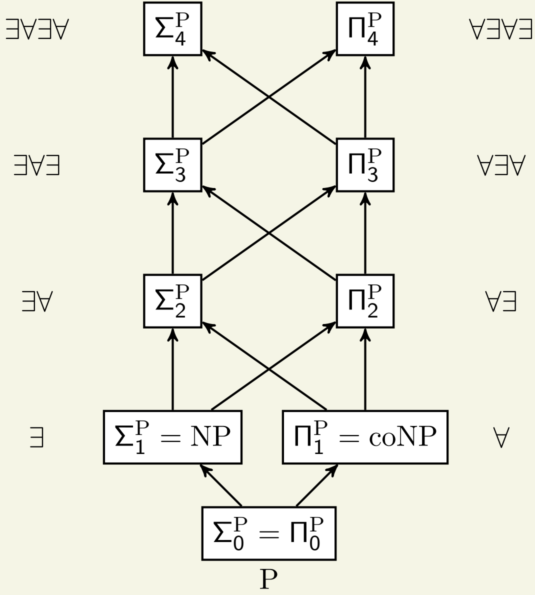
Second Level

$A \in \Sigma_2^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and polynomials p, q such that for all n and all $x \in \Sigma^n$,

$$x \in A \iff (\exists w \in \Sigma^{\leq q(n)})(\forall y \in \Sigma^{\leq p(n)})\ \langle x, w, y \rangle \in B.$$

$A \in \Pi_2^{\mathrm{P}}$ if there exist $B \in \mathrm{P}$ and polynomials p, q such that for all n and all $x \in \Sigma^n$,

$$x \in A \iff (\forall w \in \Sigma^{\leq q(n)})(\exists y \in \Sigma^{\leq p(n)})\ \langle x, w, y \rangle \in B.$$



The Polynomial-Time Hierarchy

Defining the Polynomial-Time Hierarchy

Formally, the Polynomial-Time Hierarchy is defined inductively.

Defining the Polynomial-Time Hierarchy

Formally, the Polynomial-Time Hierarchy is defined inductively.

- Define
$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}.$$

Defining the Polynomial-Time Hierarchy

Formally, the Polynomial-Time Hierarchy is defined inductively.

- Define
$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}.$$

- For $k \geq 1$, the class Σ_k^{P} consists of all B such that for some $A \in \Pi_{k-1}^{\mathrm{P}}$ and polynomial p,
$$x \in B \iff (\exists w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in A.$$

Defining the Polynomial-Time Hierarchy

Formally, the Polynomial-Time Hierarchy is defined inductively.

- Define
$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}.$$
- For $k \geq 1$, the class Σ_k^{P} consists of all B such that for some $A \in \Pi_{k-1}^{\mathrm{P}}$ and polynomial p,
$$x \in B \iff (\exists w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in A.$$
- For $k \geq 1$, define
$$\Pi_k^{\mathrm{P}} = \mathrm{co}\Sigma_k^{\mathrm{P}} = \{A \mid A^c \in \Sigma_k^{\mathrm{P}}\}.$$

Defining the Polynomial-Time Hierarchy

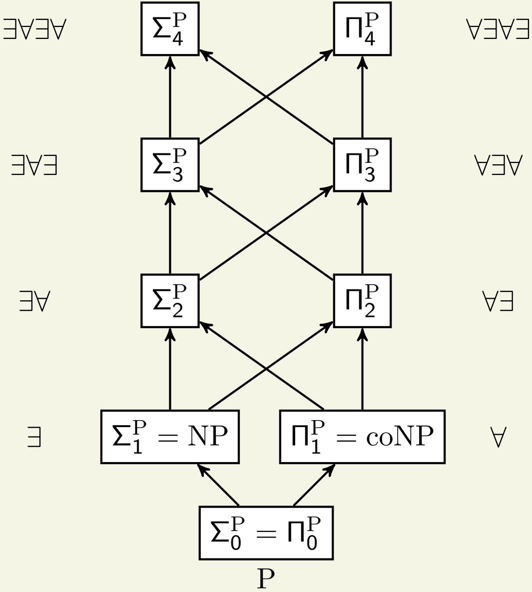
Formally, the Polynomial-Time Hierarchy is defined inductively.

- Define
$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}.$$

- For $k \geq 1$, the class Σ_k^{P} consists of all B such that for some $A \in \Pi_{k-1}^{\mathrm{P}}$ and polynomial p,
$$x \in B \iff (\exists w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in A.$$

- For $k \geq 1$, define
$$\Pi_k^{\mathrm{P}} = \mathrm{co}\Sigma_k^{\mathrm{P}} = \{A \mid A^c \in \Sigma_k^{\mathrm{P}}\}.$$

  Equivalently, Π_k^{P} consists of all B such that for some $A \in \Sigma_{k-1}^{\mathrm{P}}$ and polynomial p,
$$x \in B \iff (\forall w \in \Sigma^{\leq p(n)})\ \langle x, w \rangle \in A.$$



The Polynomial-Time Hierarchy

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

Proposition

$\text{UNIQUE-SAT} \in \Sigma_2^{\mathrm{P}}$.

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

Proposition

UNIQUE-SAT $\in \Sigma_2^{\mathrm{P}}$.

Proof.

We have

$$\begin{aligned} &\phi \in \text{UNIQUE-SAT} \\ &\iff (\exists \tau)\ \tau \text{ satisfies } \phi \text{ and } (\forall \tau')\ [\tau = \tau' \text{ or } \tau' \text{ does not satisfy } \phi] \end{aligned}$$

Unique Satisfiability Problem

$$\text{UNIQUE-SAT} = \{\phi \,|\, \phi \text{ has a unique satisfying assignment}\}$$

Proposition

$\text{UNIQUE-SAT} \in \Sigma_2^{\mathrm{P}}$.

Proof.

We have

$$\begin{aligned} &\phi \in \text{UNIQUE-SAT} \\ &\iff (\exists \tau)\ \tau \text{ satisfies } \phi \text{ and } (\forall \tau')\ [\tau = \tau' \text{ or } \tau' \text{ does not satisfy } \phi] \\ &\iff (\exists \tau)(\forall \tau')\ \tau \text{ satisfies } \phi \text{ and } [\tau = \tau' \text{ or } \tau' \text{ does not satisfy } \phi] \end{aligned}$$

Since we have $\exists\forall$ in front of a polynomial-time predicate, $\text{UNIQUE-SAT} \in \Sigma_2^{\mathrm{P}}$. $\square$

Minimal Formula Problem

Definition

Two formulas ϕ and ψ over the same set of variables are *equivalent* if $\phi(\tau) = \psi(\tau)$ for every assignment τ. We write $\phi \equiv \psi$ if ϕ and ψ are equivalent.

Minimal Formula Problem

Definition

Two formulas ϕ and ψ over the same set of variables are *equivalent* if $\phi(\tau) = \psi(\tau)$ for every assignment τ. We write $\phi \equiv \psi$ if ϕ and ψ are equivalent.

Definition

A formula ϕ is a *minimal formula* if $\phi \not\equiv \psi$ for all ψ with $|\psi| < |\phi$.

The Minimal Formula decision problem is

$$\text{MIN-FORMULA} = \{\phi \mid \phi \text{ is a minimal formula}\}.$$

Minimal Formula Problem

$$\text{MIN-FORMULA} = \{\phi \mid \phi \text{ is a minimal formula}\}.$$

Proposition

$\text{MIN-FORMULA} \in \Pi_2^{\mathrm{P}}$

Minimal Formula Problem

$$\text{MIN-FORMULA} = \{\phi \mid \phi \text{ is a minimal formula}\}.$$

Proposition

MIN-FORMULA $\in \Pi_2^{\mathrm{P}}$

Proof. We have

$$\phi \in \text{MIN-FORMULA} \iff (\forall\psi, |\psi| < |\phi|)\ \phi \not\equiv \psi$$

Minimal Formula Problem

$$\text{MIN-FORMULA} = \{\phi \mid \phi \text{ is a minimal formula}\}.$$

Proposition

$\text{MIN-FORMULA} \in \Pi_2^{\mathrm{P}}$

Proof. We have

$$\begin{aligned} \phi \in \text{MIN-FORMULA} \quad &\Longleftrightarrow \quad (\forall \psi, |\psi| < |\phi|)\ \phi \not\equiv \psi \\ &\Longleftrightarrow \quad (\forall \psi, |\psi| < |\phi|)(\exists \tau)\ \phi(\tau) \neq \psi(\tau). \end{aligned}$$

Minimal Formula Problem

$$\text{MIN-FORMULA} = \{\phi \mid \phi \text{ is a minimal formula}\}.$$

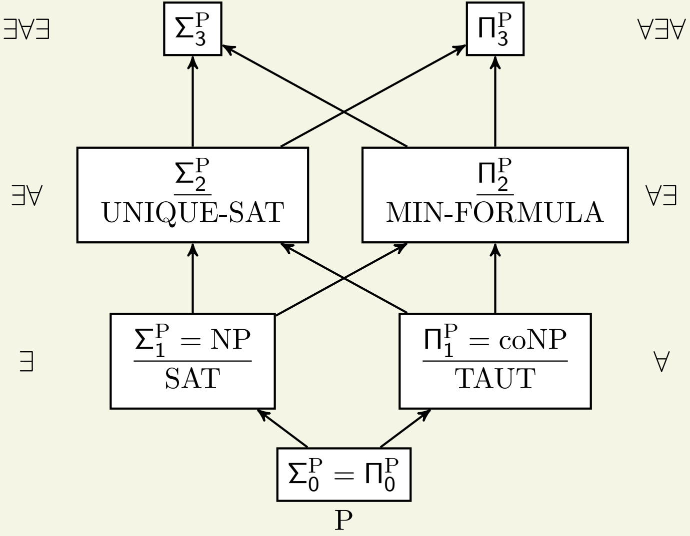
Proposition

$\text{MIN-FORMULA} \in \Pi_2^{\mathrm{P}}$

Proof. We have

$$\begin{aligned} \phi \in \text{MIN-FORMULA} &\iff (\forall\psi, |\psi| < |\phi|)\ \phi \not\equiv \psi \\ &\iff (\forall\psi, |\psi| < |\phi|)(\exists\tau)\ \phi(\tau) \neq \psi(\tau). \end{aligned}$$

Since we have $\forall\exists$ in front of a polynomial-time predicate, $\text{MIN-FORMULA} \in \Pi_2^{\mathrm{P}}$. $\square$



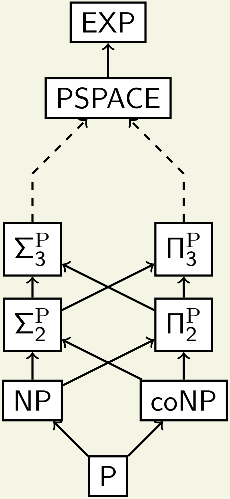
The Polynomial-Time Hierarchy

Structure of the Hierarchy

Proposition

Let $k \in \mathbb{N}$.

1. Σ_k^{P} *and* Π_k^{P} *are closed under* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-reductions.*
2. $\Sigma_k^{\mathrm{P}} = \mathrm{co}\Pi_k^{\mathrm{P}}$.
3. $\Sigma_k^{\mathrm{P}} \subseteq \Pi_{k+1}^{\mathrm{P}}$ *and* $\Pi_k^{\mathrm{P}} \subseteq \Sigma_{k+1}^{\mathrm{P}}$.
4. $\Sigma_k^{\mathrm{P}} \subseteq \Sigma_{k+1}^{\mathrm{P}}$ *and* $\Pi_k^{\mathrm{P}} \subseteq \Pi_{k+1}^{\mathrm{P}}$.
5. $\Sigma_k^{\mathrm{P}} \subseteq \mathrm{PSPACE}$ *and* $\Pi_k^{\mathrm{P}} \subseteq \mathrm{PSPACE}$.



The Polynomial-Time Hierarchy

Definition

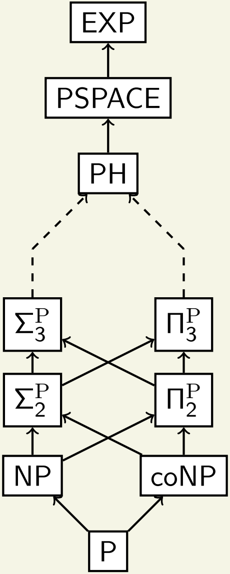
The *polynomial-time hierarchy* is

$$\mathrm{PH} = \bigcup_{k=0}^{\infty} \Sigma_k^{\mathrm{P}}.$$

Note that we can also define PH using Π_k^{P} instead of Σ_k^{P}.

Proposition

PH $\subseteq$ PSPACE.



Complete Problems for PH

A *quantified Boolean formula* (QBF) is a propositional formula preceded by quantifiers over the variables. There are no free variables. An example is

$$\phi = (\exists x_1)(\forall x_2)(\exists x_3)(x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_1).$$

Complete Problems for PH

A *quantified Boolean formula* (QBF) is a propositional formula preceded by quantifiers over the variables. There are no free variables. An example is

$$\phi = (\exists x_1)(\forall x_2)(\exists x_3)(x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_1).$$

The order in which the variables are quantified matters. You can think of it terms of a game.

- Player I picks the values for the existentially quantified variables, and Player II picks the values for the universally quantified variables.

Complete Problems for PH

A *quantified Boolean formula* (QBF) is a propositional formula preceded by quantifiers over the variables. There are no free variables. An example is

$$\phi = (\exists x_1)(\forall x_2)(\exists x_3)(x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_1).$$

The order in which the variables are quantified matters. You can think of it terms of a game.

- Player I picks the values for the existentially quantified variables, and Player II picks the values for the universally quantified variables.
- Player I's goal is to satisfy the formula; Player II's goal is to avoid satisfying the formula.

Complete Problems for PH

A *quantified Boolean formula* (QBF) is a propositional formula preceded by quantifiers over the variables. There are no free variables. An example is

$$\phi = (\exists x_1)(\forall x_2)(\exists x_3)(x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_1).$$

The order in which the variables are quantified matters. You can think of it terms of a game.

- Player I picks the values for the existentially quantified variables, and Player II picks the values for the universally quantified variables.
- Player I's goal is to satisfy the formula; Player II's goal is to avoid satisfying the formula.
- In our example, Player I picks the value for x_1. Then Player II picks a value for x_2. Lastly, Player I picks a value for x_3.

Complete Problems for PH

A *quantified Boolean formula* (QBF) is a propositional formula preceded by quantifiers over the variables. There are no free variables. An example is

$$\phi = (\exists x_1)(\forall x_2)(\exists x_3)(x_1 \vee \neg x_2) \wedge (x_3 \vee \neg x_1).$$

The order in which the variables are quantified matters. You can think of it terms of a game.

- Player I picks the values for the existentially quantified variables, and Player II picks the values for the universally quantified variables.
- Player I's goal is to satisfy the formula; Player II's goal is to avoid satisfying the formula.
- In our example, Player I picks the value for x_1. Then Player II picks a value for x_2. Lastly, Player I picks a value for x_3.
- Player I wins when he picks an x_1 such that no matter what x_2 Player II picks, he can always pick a value for x_3 that satisfies the formula.

For each $k \geq 1$, define

$$\mathrm{TQBF}_k = \left\{ \phi \,\middle|\, \begin{array}{c} \phi \text{ is a true QBF with at most } k-1 \text{ alternations} \\ \text{between } \exists \text{ and } \forall \text{, and the first quantifier is } \exists \end{array} \right\}.$$

Our example is a member of TQBF_3.

For each $k \geq 1$, define

$$\mathrm{TQBF}_k = \left\{ \phi \;\middle|\; \begin{array}{c} \phi \text{ is a true QBF with at most } k-1 \text{ alternations} \\ \text{between } \exists \text{ and } \forall \text{, and the first quantifier is } \exists \end{array} \right\}.$$

Our example is a member of TQBF_3.

Note that TQBF_1 is essentially SAT. (We just do not write out the existential quantifiers for an instance of SAT.)

For each $k \geq 1$, define

$$\mathrm{TQBF}_k = \left\{ \phi \,\middle|\, \begin{array}{c} \phi \text{ is a true QBF with at most } k-1 \text{ alternations} \\ \text{between } \exists \text{ and } \forall, \text{ and the first quantifier is } \exists \end{array} \right\}.$$

Our example is a member of TQBF_3.

Note that TQBF_1 is essentially SAT. (We just do not write out the existential quantifiers for an instance of SAT.)

Observe that for all k, $\mathrm{TQBF}_k \in \Sigma_k^{\mathrm{P}}$.

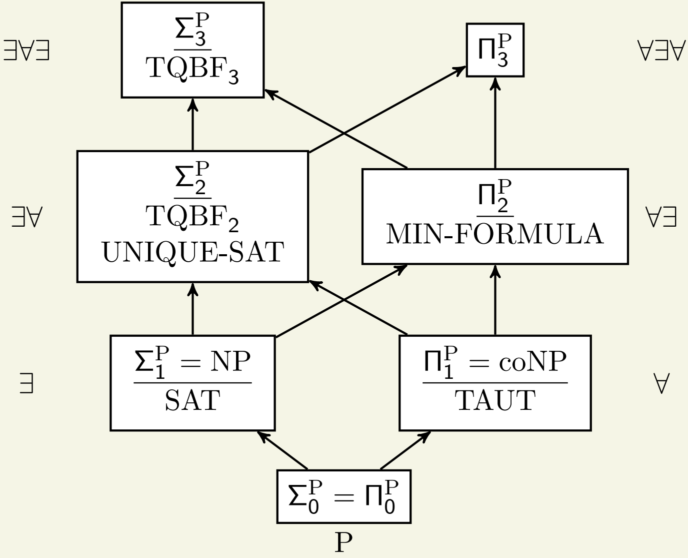
Theorem

For all $k \geq 1$, TQBF_k is Σ_k^{P}-complete.

Theorem

For all $k \geq 1$, TQBF_k is Σ_k^{P}-complete.

The proof is very similar to the proof that SAT is NP-complete.



The Polynomial-Time Hierarchy

Collapse of the Hierarchy

The polynomial-time hierarchy is an infinite collection of classes. We believe that they are all distinct but no one has a proof. For example, if $\mathrm{P} = \mathrm{PSPACE}$, then $\mathrm{P} = \mathrm{PH}$.

We say that the polynomial-time hierarchy *collapses* if there are only finitely many distinct classes. We say that it *collapses to level* k if $\mathrm{PH} = \Sigma_k^{\mathrm{P}}$.

PH Collapse Conditions

Lemma

For all $k \geq 1$, $\Sigma_k^{\mathrm{P}} = \Pi_k^{\mathrm{P}} \iff \Sigma_k^{\mathrm{P}} = \Sigma_{k+1}^{\mathrm{P}}$.

PH Collapse Conditions

Lemma

For all $k \geq 1$, $\Sigma_k^{\mathrm{P}} = \Pi_k^{\mathrm{P}} \iff \Sigma_k^{\mathrm{P}} = \Sigma_{k+1}^{\mathrm{P}}$.

Theorem

Let $k \geq 0$. *The following are equivalent.*

1. $\mathrm{PH} = \Sigma_k^{\mathrm{P}}$.
2. $\mathrm{PH} = \Pi_k^{\mathrm{P}}$.
3. $\mathrm{PH} = \Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}$.
4. $\Sigma_k^{\mathrm{P}} = \Sigma_{k+1}^{\mathrm{P}}$.

If $k \geq 1$, *these are also equivalent to*

5. $\Sigma_k^{\mathrm{P}} = \Pi_k^{\mathrm{P}}$.

PH Collapse Conditions

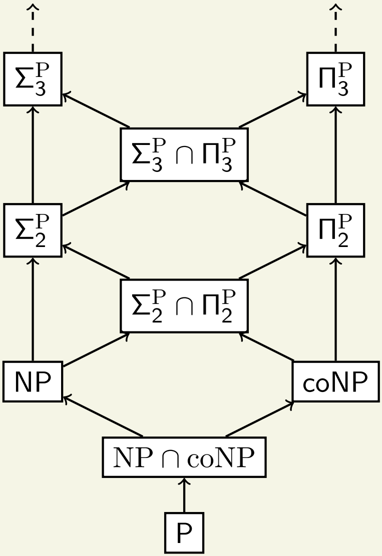
Theorem

Let $k \geq 0$. The following are equivalent.

1. $\mathrm{PH} = \Sigma_k^{\mathrm{P}}$.
2. $\mathrm{PH} = \Pi_k^{\mathrm{P}}$.
3. $\mathrm{PH} = \Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}$.
4. $\Sigma_k^{\mathrm{P}} = \Sigma_{k+1}^{\mathrm{P}}$.

If $k \geq 1$, these are also equivalent to

5. $\Sigma_k^{\mathrm{P}} = \Pi_k^{\mathrm{P}}$.



Corollary

If $\mathrm{P} = \mathrm{NP}$, *then* $\mathrm{P} = \mathrm{PH}$.

Corollary

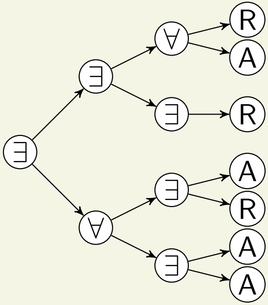
If $\mathrm{NP} = \mathrm{coNP}$, *then* $\mathrm{NP} = \mathrm{PH}$.

The intuition behind this is that since $\mathrm{NP} = \exists\mathrm{P}$ and $\mathrm{coNP} = \forall\mathrm{P}$, if the two classes are equal, then no matter how many quantifiers we have, we can replace all the universal quantifiers with existential quantifiers and collapse all the existential quantifiers to one.

Other Formulations of PH - Alternating Turing Machines

States in an alternating TM are labeled $\exists$, $\forall$, accept, or reject.

An accepting Σ_3 computation tree:



Other Formulations of PH - Alternating Turing Machines

For any $t(n)$, define the alternating time and space classes $\mathrm{ATIME}(t(n))$ and $\mathrm{ASPACE}(t(n))$.

Other Formulations of PH - Alternating Turing Machines

For any $t(n)$, define the alternating time and space classes $\mathrm{ATIME}(t(n))$ and $\mathrm{ASPACE}(t(n))$.

Define the alternating polynomial-time class

$$\mathrm{AP} = \bigcup_{c=1}^{\infty} \mathrm{ATIME}(n^c)$$

and the alternating polynomial-space class

$$\mathrm{APSPACE} = \bigcup_{c=1}^{\infty} \mathrm{ASPACE}(n^c).$$

Other Formulations of PH - Alternating Turing Machines

For any $t(n)$, define the alternating time and space classes $\mathrm{ATIME}(t(n))$ and $\mathrm{ASPACE}(t(n))$.

Define the alternating polynomial-time class

$$\mathrm{AP} = \bigcup_{c=1}^{\infty} \mathrm{ATIME}(n^c)$$

and the alternating polynomial-space class

$$\mathrm{APSPACE} = \bigcup_{c=1}^{\infty} \mathrm{ASPACE}(n^c).$$

Theorem

$$\mathrm{PSPACE} = \mathrm{AP} \textit{ and } \mathrm{EXP} = \mathrm{APSPACE}.$$

Other Formulations of PH - Alternating Turing Machines

For any $t(n)$ and $k \geq 1$, define the class

$$\Sigma_k \mathrm{TIME}(t(n))$$

of problems decidable by alternating TMs with initial state labeled $\exists$, using at most $k-1$ alternations and $O(t(n))$ time.

Theorem

For all $k \geq 1$,

$$\Sigma_k^{\mathrm{P}} = \bigcup_{c=1}^{\infty} \Sigma_k \mathrm{TIME}(n^c).$$

Other Formulations of PH - Relativization

Oracle access and relativation:

- $M^A = M$ with access to oracle A
- $\mathcal{C}^A = \mathcal{C}$ relative to oracle A
- $\mathcal{C}^{\mathcal{D}} = \mathcal{C}$ relative to $\mathcal{D}$

Other Formulations of PH - Relativization

Oracle access and relativation:

- $M^A = M$ with access to oracle A
- $\mathcal{C}^A = \mathcal{C}$ relative to oracle A
- $\mathcal{C}^{\mathcal{D}} = \mathcal{C}$ relative to $\mathcal{D}$

Theorem

$\Sigma_{k+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_k^{\mathrm{P}}}$ *for all* $k \geq 0$.

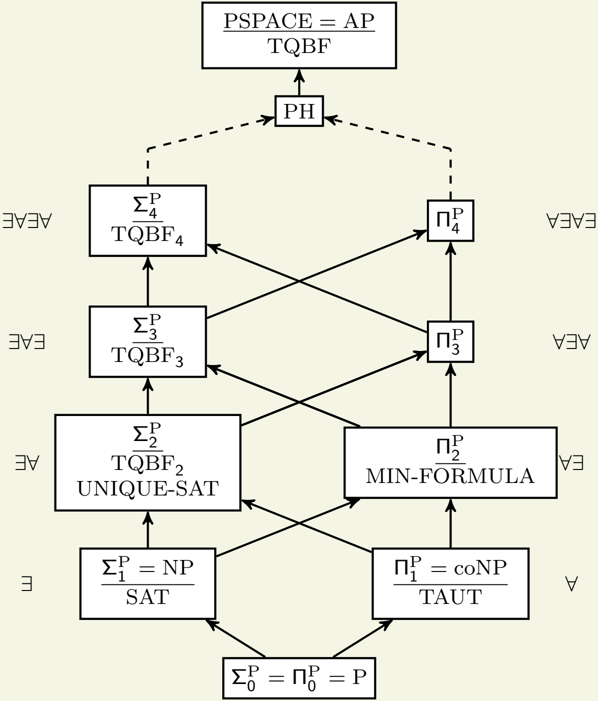
- $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$
- $\Sigma_2^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}}$
- $\Sigma_3^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}$
- $\Sigma_4^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}}$

Equivalent Formulations of PH

class	quantifier	machine	relativization
NP	$\exists\mathrm{P}$	$\mathrm{NTIME}(\mathrm{poly})$	NP
Σ_2^{P}	$\exists\forall\mathrm{P}$	$\Sigma_2\mathrm{TIME}(\mathrm{poly})$	$\mathrm{NP}^{\mathrm{NP}}$
Σ_3^{P}	$\exists\forall\exists\mathrm{P}$	$\Sigma_3\mathrm{TIME}(\mathrm{poly})$	$\mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}$

Equivalent Formulations of PH

class	quantifier	machine	relativization
NP	$\exists\mathrm{P}$	$\mathrm{NTIME}(\mathrm{poly})$	NP
Σ_2^{P}	$\exists\forall\mathrm{P}$	$\Sigma_2\mathrm{TIME}(\mathrm{poly})$	$\mathrm{NP}^{\mathrm{NP}}$
Σ_3^{P}	$\exists\forall\exists\mathrm{P}$	$\Sigma_3\mathrm{TIME}(\mathrm{poly})$	$\mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}$
coNP	$\forall\mathrm{P}$	$\mathrm{coNTIME}(\mathrm{poly})$	coNP
Π_2^{P}	$\forall\exists\mathrm{P}$	$\Pi_2\mathrm{TIME}(\mathrm{poly})$	$\mathrm{coNP}^{\mathrm{NP}}$
Π_3^{P}	$\forall\exists\forall\mathrm{P}$	$\Pi_3\mathrm{TIME}(\mathrm{poly})$	$\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$



PSPACE = AP
TQBF
PH
Σ_4^P
$TQBF_4$
Π_4^P
∃∀∃∀
∀∃∀∃
Σ_3^P
$TQBF_3$
Π_3^P
∃∀∃
∀∃∀
Σ_2^P
$TQBF_2$
UNIQUE-SAT
Π_2^P
MIN-FORMULA
∃∀
∀∃
Σ_1^P = NP
SAT
Π_1^P = coNP
TAUT
∃
∀
$\Sigma_0^P = \Pi_0^P$ = P