Computational Complexity Theory

Lecture 14: Polynomial Hierarchy

Department of Computer Science,
Indian Institute of Science

Problems between NP & PSPACE

- There are decision problems that don't appear to be captured by nondeterminism alone (i.e., with a single ∃ or ∀ quantifier), unlike problems in NP and co-NP.

- Example.

  Eq-DNF = {(ϕ,k): ϕ is a **DNF** and there's a DNF ψ of size ≤ k that is equivalent to ϕ}

- Two Boolean formulas on the same input variables are *equivalent* if their evaluations agree on every assignment to the variables.

Problems between NP & PSPACE

- There are decision problems that don't appear to be captured by nondeterminism alone (i.e., with a single ∃ or ∀ quantifier), unlike problems in NP and co-NP.

- Example.

  Eq-DNF = {(ϕ,k): ϕ is a **DNF** and there's a DNF ψ of size ≤ k that is equivalent to ϕ}

- Is Eq-DNF in NP? …if we give a DNF ψ as a certificate, it is not clear how to efficiently verify that ψ and ϕ are equivalent. (W.l.o.g. k ≤ size of ϕ .)

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Obs. Eq-DNF is in $\sum_2$.
- Proof. Think of u as another DNF ψ and v as an assignment to the variables. Poly-time TM M checks if ψ has size ≤ k and $\phi(v) = \psi(v)$.

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Obs. Eq-DNF is in $\sum_2$.
- Proof. Think of u as another DNF ψ and v as an assignment to the variables. Poly-time TM M checks if ψ has size ≤ k and $\phi(v) = \psi(v)$.

- Remark. *(Masek 1979)* Even if ϕ is given by its truth-table, the problem (i.e., DNF-MCSP) is NP-complete.

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Another example.

Succinct-SetCover = $\{(\phi_1,\ldots\phi_m,k)$: ϕ_i's are DNFs and there's an $S \subseteq [m]$ of size $\leq k$ s.t. $\bigvee_{i\in S} \phi_i$ is a tautology$\}$

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Obs. *(Homework)* Succinct-SetCover is in $\sum_2$.

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Obs. *(Homework)* Succinct-SetCover is in $\sum_2$.

- Other natural problems in PH: *"Completeness in the Polynomial-Time Hierarchy: A Compendium"* by *Schaefer and Umans (2008)*.

Class $\sum_2$

- Definition. A language L is in $\sum_2$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \iff \exists u \in \{0,1\}^{q(|x|)} \ \forall v \in \{0,1\}^{q(|x|)}$ s.t. $M(x,u,v) = 1$.

- Obs. $P \subseteq NP \subseteq \sum_2$.

Class $\sum_i$

- Definition. A language L is in $\sum_i$ if there's a polynomial function q(.) and a poly-time TM M (the "verifier") s.t.

  $x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \ \forall u_2 \in \{0,1\}^{q(|x|)} \ Q_i u_i \in \{0,1\}^{q(|x|)}$

  s.t. $M(x,u_1,\ldots,u_i) = 1$,

  where Q_i is $\exists$ or $\forall$ if i is odd or even, respectively.

- Obs. $\sum_i \subseteq \sum_{i+1}$ for every i.

Polynomial Hierarchy

- Definition. A language L is in $\sum_i$ if there's a polynomial function $q(.)$ and a poly-time TM M (the "verifier") s.t.

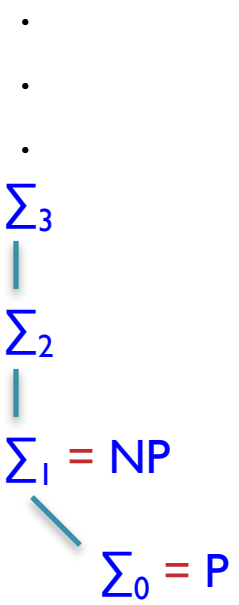
  $x \in L \Leftrightarrow \exists u_1 \in \{0,1\}^{q(|x|)} \ \forall u_2 \in \{0,1\}^{q(|x|)} \ Q_i u_i \in \{0,1\}^{q(|x|)}$

  s.t. $M(x,u_1,\ldots,u_i) = 1$,

  where Q_i is $\exists$ or $\forall$ if i is odd or even, respectively.

- Definition. *(Meyer & Stockmeyer 1972)*

$$PH = \bigcup_{i \in \mathbf{N}} \sum_i .$$



Class $\prod_i$

- Definition. $\prod_i = \text{co-}\sum_i = \{ L : \bar{L} \in \sum_i \}$.

- Obs. A language L is in $\prod_i$ if there's a polynomial function $q(.)$ and a poly-time TM M (the "verifier") s.t.

  $x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q(|x|)} \; \exists u_2 \in \{0,1\}^{q(|x|)} \; Q_i u_i \in \{0,1\}^{q(|x|)}$

  s.t. $M(x,u_1,\ldots,u_i) = 1$,

  where Q_i is $\forall$ or $\exists$ if i is odd or even, respectively.

Class $\prod_i$

- Definition. $\prod_i = \text{co-}\sum_i = \{ L : \overline{L} \in \sum_i \}$.

- Obs. A language L is in $\prod_i$ if there's a polynomial function $q(.)$ and a poly-time TM M (the "verifier") s.t.

  $x \in L \Leftrightarrow \forall u_1 \in \{0,1\}^{q(|x|)} \exists u_2 \in \{0,1\}^{q(|x|)} \; Q_i u_i \in \{0,1\}^{q(|x|)}$

  s.t. $M(x,u_1,\ldots,u_i) = 1$,

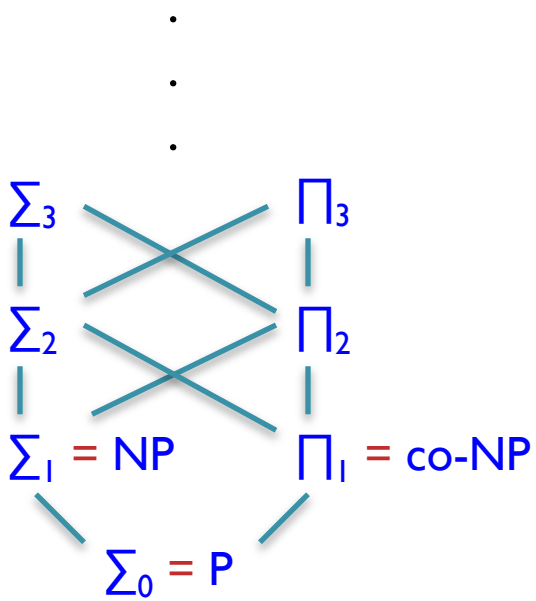
  where Q_i is $\forall$ or $\exists$ if i is odd or even, respectively.

- Obs. $\sum_i \subseteq \prod_{i+1} \subseteq \sum_{i+2}$.

Polynomial Hierarchy

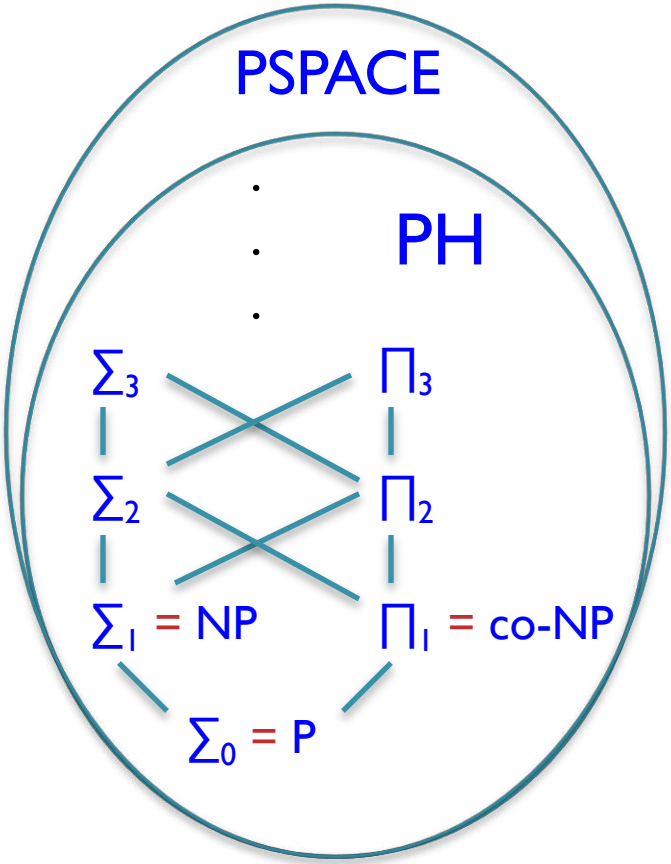
- Obs. $PH = \bigcup_{i \in \mathbf{N}} \Sigma_i = \bigcup_{i \in \mathbf{N}} \Pi_i$.

$PH =$



Polynomial Hierarchy

- Claim. PH $\subseteq$ PSPACE.
- Proof. Similar to the proof of TQBF $\in$ PSPACE.



Does PH collapse?

- General belief. Just as many of us believe P ≠ NP (i.e. $\sum_0 \neq \sum_1$) and NP ≠ co-NP (i.e. $\sum_1 \neq \prod_1$), we also believe that for every i, $\sum_i \neq \sum_{i+1}$ and $\sum_i \neq \prod_i$.

- Definition. We say PH collapses to the i-th level if $\sum_i = \sum_{i+1}$. (justified in the next theorem)

- Conjecture. There is no i such that PH collapses to the i-th level.

Does PH collapse?

- General belief. Just as many of us believe P ≠ NP (i.e. $\sum_0 \neq \sum_1$) and NP ≠ co-NP (i.e. $\sum_1 \neq \prod_1$), we also believe that for every i, $\sum_i \neq \sum_{i+1}$ and $\sum_i \neq \prod_i$.

- Definition. We say PH collapses to the i-th level if $\sum_i = \sum_{i+1}$. (justified in the next theorem)

- Conjecture. There is no i such that PH collapses to the i-th level.

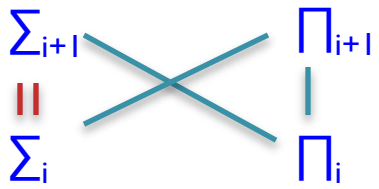
This is stronger than the P ≠ NP conjecture.

PH collapse theorems

- Theorem. If $\sum_i = \sum_{i+1}$ then PH = $\sum_i$.

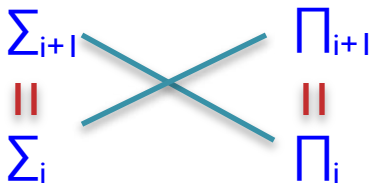
PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof.



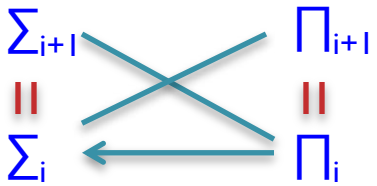
PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof.



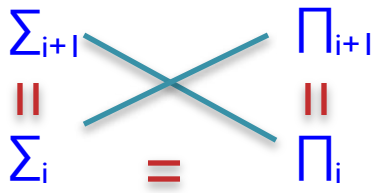
PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof.



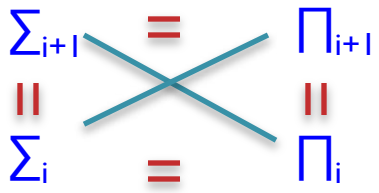
PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof.



PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof.



PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

PH collapse theorems

- Theorem. If $\sum_i = \sum_{i+1}$ then PH = $\sum_i$.
- Proof. Hence $\sum_i = \sum_{i+1} = \prod_i = \prod_{i+1}$.

  Goal is to show that $\sum_{i+1} = \sum_{i+2}$.

- Let L be a language in $\sum_{i+2}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Define L' = $\{(x, u_1)$: $\forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1\}$

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Clearly, L' is in $\Pi_{i+1} = \Sigma_i$.

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1$ s.t. $(x, u_1) \in L'$.

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1 \exists v_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, u_1, v_1 \ldots, v_i) = 1$, where N is a poly-time TM.

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1\, \exists v_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, u_1, v_1 \ldots, v_i) = 1$.

  Merge the quantifiers

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Also, $x \in L \Leftrightarrow \exists v'_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, v'_1 \ldots, v_i) = 1$.

PH collapse theorems

- Theorem. If $\Sigma_i = \Sigma_{i+1}$ then PH = Σ_i .
- Proof. Hence $\Sigma_i = \Sigma_{i+1} = \Pi_i = \Pi_{i+1}$.

  Goal is to show that $\Sigma_{i+1} = \Sigma_{i+2}$.

- Let L be a language in Σ_{i+2} . Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+2} u_{i+2}$ s.t. $M(x, u_1, \ldots, u_{i+2}) = 1$.

- Hence, L is a language in $\Sigma_i = \Sigma_{i+1}$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Define L' = $\{(x, u_1)$: $\forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1\}$

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Clearly, L' is in $\prod_i = \sum_i$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1$ s.t. $(x, u_1) \in L'$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1 \exists v_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, u_1, v_1 \ldots, v_i) = 1$, where N is a poly-time TM.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Also, $x \in L \Leftrightarrow \exists u_1\, \exists v_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, u_1, v_1 \ldots, v_i) = 1$.

  Merge the quantifiers

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Also, $x \in L \Leftrightarrow \exists v'_1 \forall v_2 \ldots Q_i v_i$ s.t. $N(x, v'_1 \ldots, v_i) = 1$.

PH collapse theorems

- Theorem. If $\sum_i = \prod_i$ then PH = $\sum_i$.
- Proof. Goal is to show that $\sum_i = \prod_i \Rightarrow \sum_i = \sum_{i+1}$.

- Let L be a language in $\sum_{i+1}$. Then there's a polynomial function q(.) and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{i+1} u_{i+1}$ s.t. $M(x, u_1, \ldots, u_{i+1}) = 1$.

- Hence, L is a language in $\sum_i$.

Complete problems in PH ?

- Recall, to define completeness of a complexity class, we need an appropriate notion of a *reduction*.
- What kind of reductions will be suitable is guided by *a complexity question*, like a comparison between the complexity class under consideration & another class.
- Is P = PH ? …use poly-time Karp reduction!

- Definition. A language L' is *PH-hard* if for every L in PH, $L \leq_p L'$. Further, if L' is in PH then L' is *PH-complete*.

Complete problems in PH ?

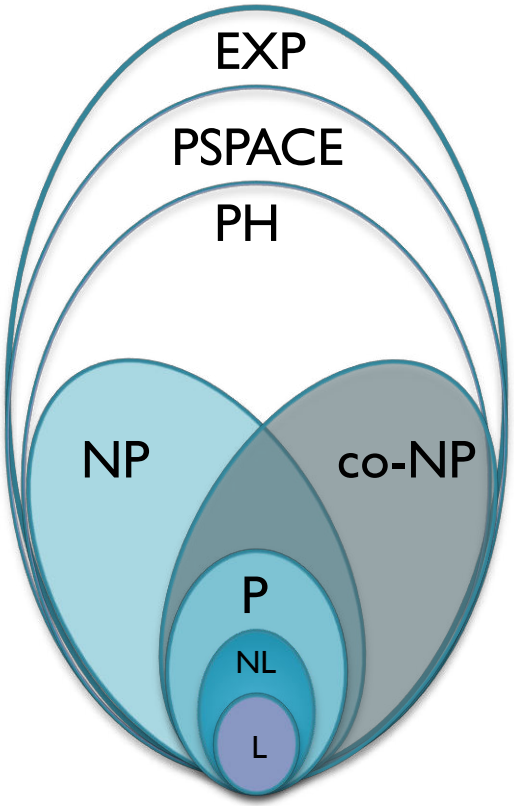
- Fact. If L is poly-time reducible to a language in $\sum_i$ then L is in $\sum_i$. (we've seen a similar fact for NP)

Complete problems in PH ?

- Fact. If L is poly-time reducible to a language in $\sum_i$ then L is in $\sum_i$. (we've seen a similar fact for NP)
- Observation. If PH has a complete problem then PH collapses.
- Proof. If L is *PH-complete* then L is in $\sum_i$ for some i. Now use the above fact to infer that PH = $\sum_i$.

Complete problems in PH ?

- Fact. If L is poly-time reducible to a language in $\sum_i$ then L is in $\sum_i$. (we've seen a similar fact for NP)
- Corollary. PH $\subsetneq$ PSPACE unless PH collapses.



Complete problems in $\sum_i$

- Recall, to define completeness of a complexity class, we need an appropriate notion of a *reduction*.
- What kind of reductions will be suitable is guided by *a complexity question*, like a comparison between the complexity class under consideration & another class.
- Is $P = \sum_i$? …use poly-time Karp reduction!

- Definition. A language L' is $\sum_i$*-hard* if for every L in $\sum_i$, $L \leq_p L'$. Further, if L' is in $\sum_i$ then L' is $\sum_i$*-complete*.

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with ∃.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*. ($\sum_1$-SAT is just SAT)

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Easy to see that $\sum_i$-SAT is in $\sum_i$.

  $x = \exists v_1 \forall v_2 \ldots Q_i v_i \; \phi(v_1, \ldots, v_i) \in \sum_i\text{-SAT}$ ⇔

  $\exists u_1 \forall u_2 \ldots Q_i u_i$ s.t. $M(x, u_1, \ldots, u_i) = 1$,

  where M outputs $\phi(u_1, \ldots, u_i)$.

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

$$x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \quad \text{s.t.} \quad M(x, u_1, \ldots, u_i) = 1.$$

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

$$x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \quad \text{s.t.} \quad \underbrace{\phi(x, u_1, \ldots, u_i)}_{\text{Boolean circuit (by Cook-Levin)}} = 1.$$

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \ \ \phi(x, u_1, \ldots, u_i)$ is true .

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \ \ \phi(x, u_1, \ldots, u_i)$ is true .

- Issue: ϕ needn't be a formula.

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

  $x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \ \ \phi(x, u_1, \ldots, u_i)$ is true .

- Observation. From the proof of the Cook-Levin theorem, we can assume that ϕ is a CNF (if i is odd) or a DNF (if i is even). *(Homework)*

Complete problems in $\sum_i$

- Definition. The language $\sum_i$-SAT contains all *true* QBF with i alternating quantifiers starting with $\exists$.

- Theorem. $\sum_i$-SAT is $\sum_i$ *-complete*.
- Proof. Let L be a language in $\sum_i$. Then there's a polynomial function $q(.)$ and a poly-time TM M s.t.

$$x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_i u_i \;\; \phi(x, u_1, \ldots, u_i) \in \sum_i\text{-SAT} .$$

Other complete problems in $\sum_2$

- Ref. *"Completeness in the Polynomial-Time Hierarchy: A Compendium"* by *Schaefer and Umans (2008)*.

- Theorem. Eq-DNF and Succinct-SetCover are $\sum_2$*-complete*.