CS151
Complexity Theory

Lecture 12

May 6, 2004

Outline

- The Polynomial-Time Hierarachy (**PH**)
- Complete problems for classes in **PH**, **PSPACE**
- **BPP** and the **PH**
- non-uniformity and the **PH**

The Polynomial-Time Hierarchy

$$\Sigma_0 = \Pi_0 = P$$

$\Delta_1 = P^P$ $\Sigma_1 = NP$ $\Pi_1 = coNP$

$\Delta_2 = P^{NP}$ $\Sigma_2 = NP^{NP}$ $\Pi_2 = coNP^{NP}$

$\Delta_{i+1} = P^{\Sigma_i}$ $\Sigma_{i+i} = NP^{\Sigma_i}$ $\Pi_{i+1} = coNP^{\Sigma_i}$

Polynomial Hierarchy **PH** $= \cup_i \Sigma_i$

The Polynomial-Time Hierarchy

$$\Sigma_0 = \Pi_0 = P$$

$$\Delta_{i+1} = P^{\Sigma_i} \quad \Sigma_{i+i} = NP^{\Sigma_i} \quad \Pi_{i+1} = coNP^{\Sigma_i}$$

- Example:
  - MIN CIRCUIT: given Boolean circuit C, integer k; is there a circuit C' of size at most k that computes the same function C does?
  - MIN CIRCUIT $\in \Sigma_2$

The Polynomial-Time Hierarchy

$$\Sigma_0 = \Pi_0 = P$$

$$\Delta_{i+1} = P^{\Sigma_i} \quad \Sigma_{i+i} = NP^{\Sigma_i} \quad \Pi_{i+1} = coNP^{\Sigma_i}$$

- Example:
  - EXACT TSP: given a weighted graph G, and in integer k; is the k-th bit of the length of the *shortest* TSP tour in G a 1?
  - EXACT TSP $\in \Delta_2$

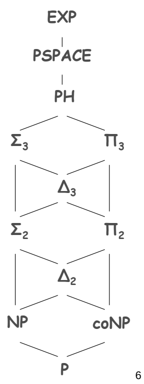
The PH

PSPACE: generalized geography, 2-person games…

3rd level: V-C dimension…

2nd level: MIN CIRCUIT, Succinct Set Cover, **BPP…**

1st level: SAT, UNSAT, factoring, etc…



Useful characterization

- Recall: $L \in$ **NP** iff expressible as
  $L = \{ x \mid \exists y, |y| \leq |x|^k, (x, y) \in R \}$
  where $R \in$ **P**.
- Corollary: $L \in$ **coNP** iff expressible as
  $L = \{ x \mid \forall y, |y| \leq |x|^k, (x, y) \in R \}$
  where $R \in$ **P**.

Useful characterization

Theorem: $L \in \mathbf{\Sigma_i}$ iff expressible as
$L = \{ x \mid \exists y, |y| \leq |x|^k, (x, y) \in R \}$
where $R \in \mathbf{\Pi_{i-1}}$.

- Corollary: $L \in \mathbf{\Pi_i}$ iff expressible as
  $L = \{ x \mid \forall y, |y| \leq |x|^k, (x, y) \in R \}$
  where $R \in \mathbf{\Sigma_{i-1}}$.

Useful characterization

- Proof of Theorem:
  - induction on i
  - base case on previous slide

  $(\Leftarrow)$
  - we know $\mathbf{\Sigma_i = NP^{\Sigma_{i-1}} = NP^{\Pi_{i-1}}}$
  - guess y, ask oracle if $(x, y) \in R$

Useful characterization

- Proof (continued):
  $(\Rightarrow)$
  - given $L \in \mathbf{\Sigma_i = NP^{\Sigma_{i-1}}}$ decided by ONTM M running in time n^k
  - try: $R = \{ (x, y) :$ y describes valid path of M's computation leading to $q_{accept} \}$
  - but how to recognize valid computation path when it depends on result of oracle queries?

Useful characterization

- Proof (continued):
  - try: $R = \{ (x, y) :$ y describes valid path of M's computation leading to $q_{accept} \}$
  - valid path = step-by-step description including **correct** yes/no answer for each A-oracle query z_j $(A \in \mathbf{\Sigma_{i-1}})$
  - verify "no" queries in $\mathbf{\Pi_{i-1}}$:
    e.g: $z_1 \notin A \wedge z_3 \notin A \wedge \ldots \wedge z_8 \notin A$
  - for each "yes" query z_j: $\exists w_j, |w_j| \leq |z_j|^k$ with $(z_j, w_j) \in R'$ for some $R' \in \mathbf{\Pi_{i-2}}$ by induction.
  - for each "yes" query z_j put w_j in description of path y

Useful characterization

- Proof (continued):
  - single language R in $\mathbf{\Pi_{i-1}}$:

    $(x, y) \in R$

    $\Leftrightarrow$

    all "no" $z_j \notin A$ and

    all "yes" z_j have $(z_j, w_j) \in R'$ and

    y is a path leading to q_{accept}.
  - Note: AND of $\mathbf{\Pi_{i-1}}$ predicates is in $\mathbf{\Pi_{i-1}}$.

Alternating quantifiers

Nicer, more usable version:

- $L \in \mathbf{\Sigma_i}$ iff expressible as

  $L = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q y_i\, (x, y_1, y_2, \ldots, y_i) \in R \}$

  where $Q = \forall/\exists$ if i even/odd, and $R \in \mathbf{P}$

- $L \in \mathbf{\Pi_i}$ iff expressible as

  $L = \{ x \mid \forall y_1 \exists y_2 \forall y_3 \ldots Q y_i\, (x, y_1, y_2, \ldots, y_i) \in R \}$

  where $Q = \exists/\forall$ if i even/odd, and $R \in \mathbf{P}$

Alternating quantifiers

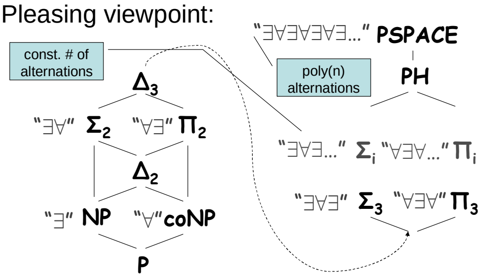
- Proof:
  - ($\Rightarrow$) induction on i
  - base case: true for $\mathbf{\Sigma_1}$**=NP** and $\mathbf{\Pi_1}$**=coNP**
  - consider $L \in \mathbf{\Sigma_i}$:

    $L = \{x \mid \exists y_1\, (x, y_1) \in R' \}$, for $R' \in \mathbf{\Pi_{i-1}}$

    $L = \{x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q y_i\, (x, y_1, y_2, \ldots, y_i) \in R\}$
  - same argument for $L \in \mathbf{\Pi_i}$
  - ($\Leftarrow$) exercise.

Alternating quantifiers

Pleasing viewpoint:



Complete problems

- Recall:

MIN CIRCUIT: given Boolean circuit C, integer k; is there a circuit C' of size at most k that computes the same function C does?

$\{ (C, k) \mid \exists C'\, \forall x\, (|C'| \leq k \text{ and } C'(x) = C(x)) \}$

- Conclude: in $\mathbf{\Sigma_2}$
- (open whether it is complete for $\mathbf{\Sigma_2}$)

Complete problems

- three variants of SAT:
  - QSAT_i (i odd) =

    {3-CNFs $\varphi(x_1, x_2, \ldots, x_i)$ for which

    $\exists x_1 \forall x_2 \exists x_3 \ldots \exists x_i\, \varphi(x_1, x_2, \ldots, x_i) = 1$}
  - QSAT_i (i even) =

    {3-DNFs $\varphi(x_1, x_2, \ldots, x_i)$ for which

    $\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_i\, \varphi(x_1, x_2, \ldots, x_i) = 1$ }
  - QSAT = {3-CNFs φ for which

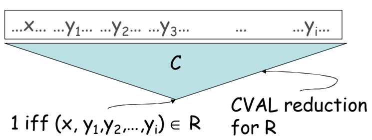
    $\exists x_1 \forall x_2 \exists x_3 \ldots Q x_n\, \varphi(x_1, x_2, \ldots, x_n) = 1$}

QSAT_i is $\mathbf{\Sigma_i}$-complete

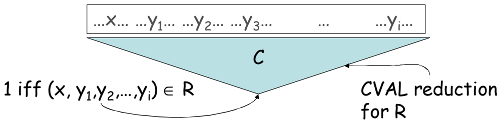
Theorem: QSAT_i is $\mathbf{\Sigma_i}$-complete.

- Proof: (clearly in $\mathbf{\Sigma_i}$)
  - assume i odd; given $L \in \mathbf{\Sigma_i}$ in form

    $\{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \exists y_i\, (x, y_1, y_2, \ldots, y_i) \in R \}$



QSAT_i is $\boldsymbol{\Sigma_i}$-complete



- Problem set: can construct 3-CNF φ from C:

  $\exists z\ \varphi(x,y_1,\ldots,y_i,z) = 1 \Leftrightarrow C(x,y_1,\ldots,y_i) = 1$
- we get:

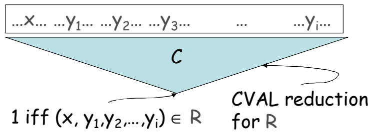
  $\exists y_1\forall y_2\ldots\exists y_i\exists z\ \varphi(x,y_1,\ldots,y_i,z) = 1$

  $\Leftrightarrow \exists y_1\forall y_2\ldots\exists y_i C(x,y_1,\ldots,y_i) = 1 \Leftrightarrow x \in L$

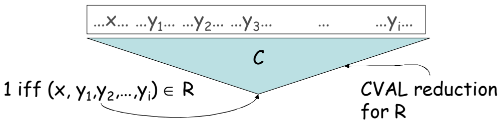
QSAT_i is $\boldsymbol{\Sigma_i}$-complete

- Proof (continued)
  - assume i even; given $L \in \boldsymbol{\Sigma_i}$ in form

    $\{ x \mid \exists y_1\forall y_2\exists y_3 \ldots \forall y_i\ (x, y_1,y_2,\ldots,y_i) \in R \}$



QSAT_i is $\boldsymbol{\Sigma_i}$-complete



- Problem set: can construct 3-DNF φ from C:

  $\forall z\ \varphi(x,y_1,\ldots,y_i,z) = 1 \Leftrightarrow C(x,y_1,\ldots,y_i) = 1$
- we get:

  $\exists y_1\forall y_2\ldots\forall y_i\forall z\ \varphi(x,y_1,y_2,\ldots,y_i,z) = 1$

  $\Leftrightarrow \exists y_1\forall y_2\ldots\forall y_i C(x,y_1,y_2,\ldots,y_i) = 1 \Leftrightarrow x \in L$

QSAT is **PSPACE**-complete

Theorem: QSAT is **PSPACE**-complete.

- Proof:
  - in **PSPACE**: $\exists x_1\forall x_2\exists x_3 \ldots Qx_n\ \varphi(x_1, x_2, \ldots, x_n)$?
  - "$\exists x_1$": for each x_1, recursively solve

    $\forall x_2\exists x_3 \ldots Qx_n\ \varphi(x_1, x_2, \ldots, x_n)$?
    - if encounter "yes", return "yes"
  - "$\forall x_1$": for each x_1, recursively solve

    $\exists x_2\forall x_3 \ldots Qx_n\ \varphi(x_1, x_2, \ldots, x_n)$?
    - if encounter "no", return "no"
  - base case: evaluating a 3-CNF expression
  - poly(n) recursion depth
  - poly(n) bits of state at each level

QSAT is **PSPACE**-complete

- Proof (continued):
  - given TM M deciding $L \in$ **PSPACE**; input x
  - configuration graph has 2^{n^k} nodes
  - recall:

    PATH(X, Y, i) ⇔ path from X to Y of length at most 2^i
  - goal: 3-CNF $\varphi(w_1,w_2,w_3,\ldots,w_m)$

    $\exists w_1\forall w_2\ldots Qw_m\ \varphi(w_1,\ldots,w_m)$

    $\Leftrightarrow$ PATH(START, ACCEPT, n^k)

QSAT is **PSPACE**-complete

- for i = 0, 1, … n^k produce quantified Boolean **expressions** $\psi_i(A, B)$

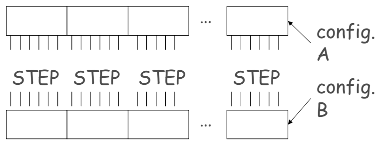
  $\exists w_1\forall w_2\ldots\ \psi_i(A, B, W) \Leftrightarrow \text{PATH}(A, B, i)$
- convert ψ_{n^k} to 3-CNF φ
  - add variables V
- hardwire START, ACCEPT

  $\exists w_1\forall w_2\ldots\ \exists V\ \varphi(W, V) \Leftrightarrow x \in L$

QSAT is **PSPACE**-complete

- Proof (continued):
  - $\psi_o(A, B) = 1$ iff
    - A = B or
    - A yields B in one step of M

    Boolean expression of size $O(n^k)$



QSAT is **PSPACE**-complete

- recall Savitch's algorithm:

$$\text{PATH}(A, B, i+1)$$
$$\Leftrightarrow$$
$$\exists Z\,[\text{PATH}(A, Z, i) \wedge \text{PATH}(Z, B, i)]$$

- cannot define $\psi_{i+1}(A, B)$ to be

$$\exists Z\,[\psi_i(A, Z) \wedge \psi_i(Z, B)]$$

  (why?)

QSAT is **PSPACE**-complete

- Proof (continued):
  - Key: reuse expressions just as Savitch reuses stack records…
  - define $\psi_{i+1}(A, B)$ to be

$$\exists Z \forall X \forall Y\,[((X=A \wedge Y=Z) \vee (X=Z \wedge Y=B)) \Rightarrow \psi_i(X, Y)]$$

  - $\psi_i(X, Y)$ is preceded by quantifiers
  - move to front (they don't involve X,Y,Z,A,B)

QSAT is **PSPACE**-complete

$\psi_o(A, B) = 1$ iff A = B or A yields B in 1 step

$$\exists Z \forall X \forall Y\,[((X=A \wedge Y=Z) \vee (X=Z \wedge Y=B)) \Rightarrow \psi_i(X, Y)]$$

- $|\psi_0| = O(n^k)$
- $|\psi_{i+1}| = O(n^k) + |\psi_i|$

- total size of ψ_{nk} is $O(n^k)^2$ = poly(n)
- logspace reduction

PH collapse

Theorem: if $\mathbf{\Sigma_i = \Pi_i}$ then for all j > i

$$\mathbf{\Sigma_j = \Pi_j = \Delta_j = \Sigma_i}$$

"the polynomial hierarchy collapses to the i-th level"

- Proof:
  - sufficient to show $\mathbf{\Sigma_i = \Sigma_{i+1}}$
  - then $\mathbf{\Sigma_{i+1} = \Sigma_i = \Pi_i = \Pi_{i+1}}$; apply theorem again

PH collapse

- recall: $L \in \mathbf{\Sigma_{i+1}}$ iff expressible as

$$L = \{ x \mid \exists\, y\ (x, y) \in R \}$$

  where $R \in \mathbf{\Pi_i}$
- since $\mathbf{\Pi_i = \Sigma_i}$, R expressible as

$$R = \{ (x,y) \mid \exists\, z\ ((x, y), z) \in R' \}$$

  where $R' \in \mathbf{\Pi_{i-1}}$
- together: $L = \{ x \mid \exists\, (y,z)\ \ (x, (y,z)) \in R' \}$
- conclude $L \in \mathbf{\Sigma_i}$

Oracles vs. Algorithms

A point to ponder:

- given poly-time **algorithm** for SAT
  - can you solve MIN CIRCUIT efficiently?
  - what other problems? Entire complexity classes?
- given SAT **oracle**
  - same input/output behavior
  - can you solve MIN CIRCUIT efficiently?

Natural complete problems

- We now have versions of SAT complete for levels in **PH**, **PSPACE**
- Natural complete problems?
  - **PSPACE**: games
  - **PH**: almost all natural problems lie in the second level

Natural complete problems

- MIN CIRCUIT
  - good candidate, still open
- MIN DNF: given DNF φ, integer k; is there a DNF φ' of size at most k computing same function φ does?
- example:

$$x_1x_2x_3 \vee x_1x_2\neg x_3 \vee x_4$$

Natural complete problems

- MIN CIRCUIT
  - good candidate, still open
- MIN DNF: given DNF φ, integer k; is there a DNF φ' of size at most k computing same function φ does?
- example:

$$x_1x_2x_3 \vee x_1x_2\neg x_3 \vee x_4 \equiv x_1x_2 \vee x_4$$

Simpler version of MIN DNF

Theorem (U): MIN DNF is $\mathbf{\Sigma_2}$-complete.

- we'll consider a simpler variant:
  - IRREDUNDANT: given DNF φ, integer k; is there a DNF φ' consisting of at most k terms of φ computing same function φ does?

Simpler version of MIN DNF

- analogy with an **NP**-complete problem:
  - SET COVER: given subsets $S_1, S_2, \ldots, S_m \subset U$, integer k, is there a collection of at most k sets that cover U.