Tuesday, May 8, 2018 2:52 PM

Oracle Turing Machine (OTM)

multi-tape TM with a special "query" tape.

Special States $q_?$ q_{yes} q_{no}.

On input x w/ oracle language A

M_A runs as usual except

if M_A enters $q_?$

y = contents of query tape.

$y \in A \Rightarrow$ transitions to q_{yes}.

$y \notin A \Rightarrow$ transitions to q_{no}

$\text{co-NP} \subseteq P^{SAT}$

Non-deterministic OTM: defined the same way

transition is a relation instead of a function.

Oracle is like a subroutine

- Each call counts as only one step.

Poly-time OTM with a SAT oracle can solve

given $\phi_1\ \phi_2 \cdots \phi_n$ are an even # satisfiable?

Shorthand: applying an oracle to an entire complexity class:

C - complexity class A - language.

$C^A = \{ L :$ decidable by OTM M w/ oracle A with M in $C \}$.

Example P^{SAT}

Tuesday, May 8, 2018 9:10 PM

$\text{co-NP} \not\subseteq \text{NP}$

$\text{co-NP} \subseteq P^{SAT}$.

Another shorthand: using a complexity class as an oracle.

OTM M Complexity Class $\mathcal{C}$

$M^{\mathcal{C}}$ decides L if for some language $A \in \mathcal{C}$ M^A decides L.

Both together: $\mathcal{C}^{\mathcal{D}}$ = languages decidable by OTM in $\mathcal{C}$ w/ oracle language in $\mathcal{D}$.

$P^{SAT} = P^{NP}$

We can use these definitions to define new complexity classes

- which ones have complete problems?
- have natural interpretation using first-order logic.
- help state consequences.

$\Sigma_0 = \Pi_0 = P$

$\Delta_1 = P^P \qquad \Sigma_1 = NP \qquad \Pi_1 = \text{co-NP}.$

$\Delta_2 = P^{NP} \qquad \Sigma_2 = NP^{NP} \qquad \Pi_2 = \text{co-NP}^{NP}$

$\vdots$

$\Delta_{i+1} = P^{\Sigma_i} \qquad \Sigma_{i+1} = NP^{\Sigma_i} \qquad \Pi_{i+1} = \text{co-NP}^{\Sigma_i}$

Polynomial Hierarchy: $PH = \bigcup_i \Sigma_i$

Tuesday, May 8, 2018 9:17 PM

NP^{NP} NP^{SAT}.

Examples: MINCIRCUIT $\in \Sigma_2$.

Input: (C, s) circuit, integer

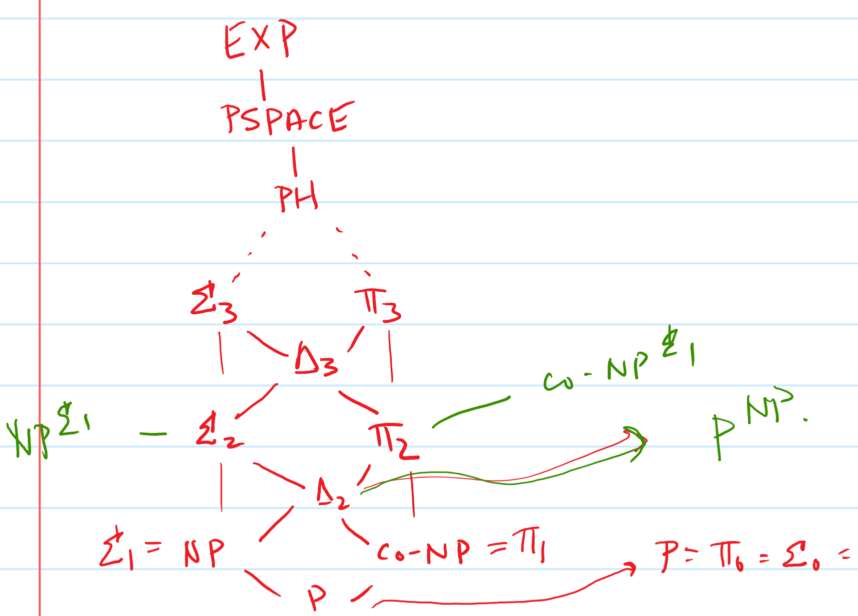
Is there a circuit w/ $\leq s$ gates that computes the same function as C?

Do C + C' differ on some input? $\in$ NP.
Do C + C' compute the same function $\in$ co-NP

Guess C' with $\leq s$ gates, then consult oracle on equivalence.

EXACT TSP: given a weighted graph G + integer k, is the k^{th} bit of the description of the shortest TSP tour $= 1$?

EXACT TSP $\in \Delta_2 = P^{NP}$ (Binary search on TSP length).



Wednesday, May 9, 2018 9:28 AM

Theorem: • $L \in \Sigma_i$ if it is expressable as:

$$L = \{x \mid \exists y \; |y| \le |x|^k \; (x,y) \in R, \; R \in \Pi_{i-1}\}$$

• $L \in \Pi_i$ if it is expressable as:

$$L = \{x \mid \forall y \; |y| \le |x|^k \; (x,y) \in R \; \; R \in \Sigma_{i-1}\}$$

Nicer more usable version!

$L \in \Sigma_i$ if it is expressable as:

$$L = \{x \mid \exists \vec{y}_1 \forall \vec{y}_2 \exists \vec{y}_3 \cdots Q y_i \; (x, y_1, \ldots, y_i) \in R\}$$

$R \in P$.

if i odd $Q = \exists$
if i even $Q = \forall$.

$L \in \Pi_i$ if it is expressable as:

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

$R \in P$

if i is odd: $Q = \forall$
if i is even: $Q = \exists$.

Note that if $L \in \Sigma_i$ then $L \in NP^{\Sigma_{i-1}} = NP^{\Pi_{i-1}}$

$L \in \Sigma_{i-1} \iff co\text{-}L \in \Pi_{i-1}$: Can always reverse output of the oracle.

Thursday, May 10, 2018 8:50 AM

By induction on i:

$i = 1$ $\Sigma_1 = NP$

$L \in NP \iff \exists k$, polytime R.

$L = \{x \mid \exists y \ |y| \le |x|^k \ R(x,y) = yes\}$

Assume $L \in \Sigma_{i-1} \iff \exists k$, Polytime R

$L = \{x \mid \exists y_1 \forall y_2 \cdots Q y_{i-1}, |y_j| \le |x|^k$
$R(x, y_1, .., y_{i-1}) = yes\}$

Note that
if $x \notin L$ then $\forall y_1 \exists y_2 \cdots \bar{Q} y_{i-1} \ R(x, y_1, .., y_{i-1}) = no$

Therefore $co\text{-}L = \{x \mid \forall y_1 \exists y_2 \cdots \bar{Q} y_{i-1} \ \bar{R}(x, y_1, .., y_{i-1}) = yes\}$

Now consider $L \in \Sigma_i = NP^{\Sigma_{i-1}}$

This NP machine gets many queries to Σ_{i-1}, but we need to compress them into a single query.

There is a poly-time NTM
w/ oracle $A \in \Sigma_{i-1}$
Call this machine M_A

$x \in L \iff M_A$ accepts x.

Now Suppose we guess:

- M_A's non-deterministic choices y
- The inputs of the queries made to A. $z_1 \cdots z_k$
- The outcomes of those queries $u_1 \cdots u_k$

If the guesses are correct then:

$$\overbrace{}^{A \in \Sigma_{i-1}}$$

$$U_j = 1 \Rightarrow z_j \in A \Rightarrow \exists y_1 \forall y_2 \cdots Q y_{i-1} \; R(z_j, y_1, .., y_{i-1}) = yes$$

$$V_j = 0 \Rightarrow z_j \notin A \Rightarrow \forall y_1 \exists y_2 \cdots \overline{Q} y_{i-1} \; R(z_j, y_1, .., y_{i-1}) = no$$

$x \in L$ iff there is a good set of choices:

$\exists \; y \; z_1 \cdots z_k \; U_1 \cdots U_k$ [M_A w/ non-det choices y
asks queries $z_1 \cdots z_k$
gets answers $U_1 \cdots U_k$
accepts

poly time checkable.

AND for $U_j = 1$ $\exists y_1 \forall y_2 \cdots Q y_{i-1} \; R(z_j, y_1 .. y_{i-1}) = yes$
AND for $U_j = 0$ $\underline{\forall y_1} \exists y_2 \cdots \overline{Q} y_{i-1} \; R(z_j, y_1, .., y_{i-1}) = no$.]

the additional layer of alternation comes from this term.

We can rearrange the above expression to look like:

$$\{x \mid \exists y_1 \forall y_2 \cdots Q y_i \; R(x, y_1, .., y_i) = yes\}.$$

Thursday, May 10, 2018 8:50 AM

$x \in L \iff$

$\exists \underbrace{y\, z_1 \cdots z_k\ v_1 \cdots v_k}_{y_0}$ [M_A w/ non-det choices y asks queries $z_1 \cdots z_k$ gets answers $v_1 \cdots v_k$ accepts — P – poly time checkable.

AND for $v_j = 1$ $\exists y_1 \forall y_2 \cdots Q\, y_{i-1}\ R(z_j, y_1 \ldots y_{i-1}) = \text{yes}$

AND for $v_j = 0$ $\forall y_1 \exists y_2 \cdots \bar{Q}\, y_{i-1}\ R(z_j, y_1, \ldots, y_{i-1}) = \text{no}$.]

↳ the additional layer of alternation comes from this term.

$z_1 \in A$?

$$\exists y_0 \Big[P(y_0) \wedge \exists y_1 \forall y_2\, R(y_0, y_1, y_2) \wedge \exists z_1 \forall z_2\, R(y_0, z_1, z_2) \wedge \forall s_1 \exists s_2\, \neg R(y_0, s_1, s_2) \wedge \forall r_1 \exists r_2\, \neg R(y_0, r_1, r_2) \Big]$$

z_3 z_4

$$\exists y_0, y_1, z_1\ \forall y_2, z_2, s_1, r_1\ \exists s_2 r_2\ \big(P(y_0) \wedge R(y_0, y_1, y_2) \wedge R(y_0, z_1, z_2) \wedge \neg R(y_0, s_1, s_2) \wedge \neg R(y_0, r_1, r_2) \big)$$

Thursday, May 10, 2018 9:30 AM

$\Leftarrow$ $L = \{x \mid \exists y_1 \forall y_2 \cdots Q y_i \; R(x, y_1, \ldots, y_i) = \text{yes}\}$.

Show $L \in NP^{\Sigma_{i-1}}$

Consider language $L' = \{(x, y_1) \mid \forall y_2 \cdots Q y_i \; R(x, y_1, \ldots, y_i) = \text{yes}\}$

$L' \in \Pi_{i-1} \Rightarrow \text{co-}L' \in \Sigma_{i-1}$

$L = \{x \mid \exists y_1 \; (x, y_1) \in L'\}$

An NP machine w/ oracle co-L'

On input x:

Guess y_1

Query whether $(x, y_1) \in$ co-L' (L')

if oracle answers 'yes' $\Rightarrow$ reject (accept)

if oracle answers 'no' $\Rightarrow$ accept (reject).

PSPACE : $\exists x_1 \forall x_2 \cdots \; \phi(x_1, x_2, \ldots, x_n)$

poly # of alternations : number of iterations can depend on the input.

PH : every constant # of alternations.