3CNF = $(x_1 \vee x_2 \vee \neg x_7) \wedge (x_3 \vee \neg x_4 \vee x_5) \wedge -$

3DNF = $(x_1 \wedge x_2 \wedge \neg x_7) \vee (x_3 \wedge \neg x_4 \wedge x_5) \cdots$

ϕ = 3CNF

$\neg\phi$ = 3DNF.

Polynomial Hierarchy: complete problems.

Three variations of SAT

$QSAT_i$ (i odd) $\{$ 3-CNFs $\phi(\vec{x}_1, \ldots, \vec{x}_i)$ for which $\exists \vec{x}_1 \forall \vec{x}_2 \cdots \exists \vec{x}_i \; \phi(\vec{x}_1, \ldots, \vec{x}_i) = 1 \}$

$QSAT_i$ (i even) $\{$ 3-DNFs $\phi(\vec{x}_1, \ldots, \vec{x}_i)$ for which $\exists \vec{x}_1 \forall \vec{x}_2 \cdots \forall \vec{x}_i \; \phi(\vec{x}_1, \ldots, \vec{x}_i) = 1 \}$

QSAT $\{$ 3-CNFs ϕ for which $\exists x_1 \forall x_2 \cdots Q_n \; \phi(x_1, \ldots, x_n) = 1 \}$

Theorem: $QSAT_i$ is Σ_i-complete.

$QSAT_i \in \Sigma_i$

$QSAT_i = \{ \phi(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_i) \mid \exists \vec{y}_1 \forall \vec{y}_2 \cdots Q\vec{y}_i$

Thursday, May 10, 2018 9:41 AM

$\text{BoolEval}(\phi, \vec{y}_1, \vec{y}_2, \ldots, \vec{y}_i)$ accepts$\}$.

$\text{BoolEval}(\phi, \vec{y}_1, \ldots, \vec{y}_i) \iff \phi(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_i)$ evaluates to true.

Mon May 14,2018 - page 2

Friday, May 11, 2018 8:40 AM

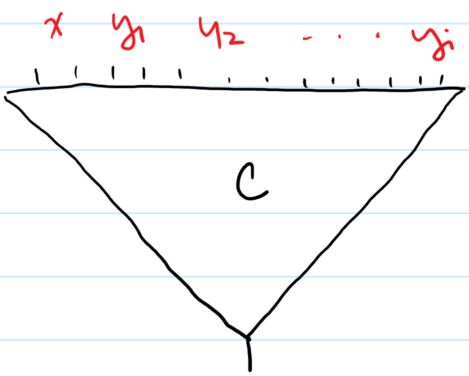
Now show $\forall\ L \in \Sigma_i \quad L \propto QSAT_i$

Assume i is odd $\quad L \in \Sigma_i$ can be expressed as:

$$\{x \mid \exists y_1 \forall y_2 \dots \exists y_i\ \underline{R(x, \vec{y}_1, \dots, \vec{y}_i)} \text{ accepts}\}$$

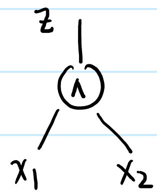
If $x, y_1, \dots, y_i$ are fixed we have polytime TM that decides R.

Tableau for R w/ input variables $x, y_1, \dots, y_i$
$\Rightarrow$ poly-sized circuit.



1 iff $R(x\, y_1 \dots y_i)$ accepts

Use the circuit C to construct a CNF formula adding some additional variables:



$(z \Leftrightarrow x_1 \wedge x_2)$

$\neg z \vee (x_1 \wedge x_2)$
$= (\neg z \vee x_1) \wedge \neg z \vee x_2)$

$z \vee \neg(x_1 \wedge x_2) = z \vee \neg x_1 \vee \neg x_2$

New clauses

Also add clause (z_n) final output.

$$\exists \vec{z}\ \phi(x, y_1, y_2, \dots, y_i, z) \Longleftrightarrow C(x, y_1, \dots, y_i) = 1.$$

Mon May 14,2018 - page 3

Friday, May 11, 2018 8:54 AM

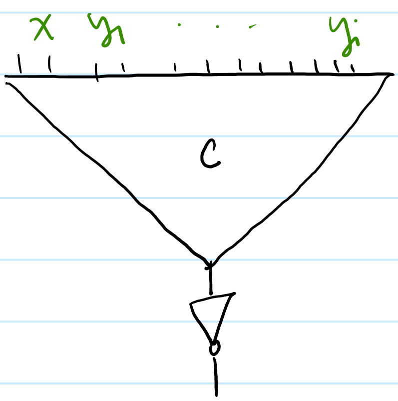
$\exists \vec{z}\ \phi(x, y_1, y_2, \ldots, y_i) \iff C(x, y_1, \ldots, y_i) = 1.$

So: $x \in L \iff \exists y_1 \forall y_2 \cdots \exists y_i\ \underbrace{C(x, y_1, \ldots, y_i) = 1}$

$\iff \exists y_1 \forall y_2 \cdots \exists y_i \exists z\ \underbrace{\phi(x, y_1, \ldots, y_i, z) = \text{true}}$

Now suppose i is even:

$L = \{x \mid \exists y_1 \forall y_2 \cdots \forall y_i\ R(x, y_1, \ldots, y_i) \text{ accepts}\}$



$\implies$ convert to CNF formula $\phi(x, y_1, \ldots, y_i, z)$ as before.

$= 0$ if $R(x, y_1, \ldots, y_i)$ accepts.

For fixed $x\ y_1 \cdots y_i$: $C(x, y_1, \ldots, y_i) = 0 \iff \forall z\ \overbrace{\phi(x, y_1, \ldots, y_i, z)}^{\text{3CNF}} = 0$

$\iff \forall z\ \underbrace{\neg\phi(x, y_1, \ldots, y_i, z) = 1}$

By De Morgan's laws this becomes 3-DNF

$x \in L \iff \exists y_1 \forall y_2 \cdots \forall y_i\ C(x, y_1, \ldots, y_i) = 0$

$\iff \exists y_1 \forall y_2 \cdots \forall y_i \forall z\ \underbrace{\phi'(x, y_1, \ldots, y_i, z) = 1}_{\phi' = \neg\phi}.$

Mon May 14,2018 - page 4

Monday, May 14, 2018 7:49 AM

QSAT is PSPACE-complete.

$\forall x_1 \exists x_2 \forall x_3 \ldots Q x_n \, \phi(x_1, x_2, \ldots, x_n)$
$\exists x_1 \forall x_2 \exists x_3 \ldots Q x_n \, \phi(x_1, x_2, \ldots, x_n)$.

QSAT $\in$ PSPACE

$x_1 = 0$ $x_2 = 1$ $x_3 = 0$
$\exists$ $\forall$ $\exists$
NO.

$\exists x_1 \cdots Q x_n$.
For each value for x_1 recursively solve
$\forall x_2 \cdots Q x_n \, \phi(x_1, \ldots, x_n)$
if "yes" then return "yes"
Return "no".

$\forall x_1 \cdots Q x_n$
For each value for x_1, recursively solve.
$\exists x_1 \forall x_2 \cdots Q x_n \, \phi(x_1, x_2, \ldots, x_n)$
If "no" return "no".
Return "yes".

Base Case: 3 CNF expression w/ all variables determined (CVAL)
poly(n) recursion depth.
poly(n) bits of state @ each level.
current value for x_1

Now for each $L \in$ PSPACE will show $L \propto$ QSAT.

Mon May 14,2018 - page 5

Monday, May 14, 2018 7:49 AM

$L \in PSPACE \Rightarrow \exists$ poly-space TM M that decides L.

$\underline{\underline{\exists k}}$: On input x : $|x| = n$, M has $\leq 2^{n^k}$ possible configurations.

$\sigma_1 \sigma_2 \cdots \sigma_j \, q \, \sigma_{j+1} \cdots \sigma_{n^k}$. Configuration can be expressed as a string.

Single start state: $q_{start} \, x_1 \cdots x_n$

Can alter M so that it has a single accept state: q_{acc}.

Just before M is about to accept, erase tape contents & move head left.

Define $\underline{REACH(X, Y, i)} \Leftrightarrow$ true iff config Y is reachable from X in $\leq 2^i$ steps.

Produce 3-CNF $\phi(w_1, w_2, \ldots, w_m)$ s.t.

$\exists w_1 \forall w_2 \cdots Q w_n \; \phi(w_1, w_2, \ldots, w_m) \Leftrightarrow REACH(start, acc, n^k)$

Will recursively define

$\underline{\psi_i(A, B)} = \exists w_1 \forall w_2 \cdots Q w_i \; \phi(A, B, w_1, \ldots, w_i)$

$\Leftrightarrow REACH(A, B, i)$.

$\phi_0 = \psi_0(A, B) =$ true iff $A = B$ or A yields B in one step of the TM.

Configs A & B are defined in terms of a poly # boolean vars.

Monday, May 14, 2018 7:49 AM

2^{i+1} 2^i 2^i

Key idea: $REACH(A,B,i+1) \iff \exists z\ [REACH(A,z,i) \wedge REACH(z,B,i)]$

this would get exponentially large!

Can not do: $\varphi_{i+1}(A,B) = \exists z\ [\varphi_i(A,z) \wedge \varphi_i(z,B)]$

Instead:

$$\varphi_{i+1}(A,B) = \exists z\ \forall x\ \forall y\ [[(X=A \wedge Y=z) \vee (X=z \wedge Y=B)] \Rightarrow \varphi_i(X,Y)]$$

A, z
z, B

Note that φ_i has quantifiers, but they don't bind A, B, X, Y, or Z so they can be moved to the front.

$|\varphi_0| = O(n^k)$.

$|\varphi_{i+1}| = O(n^k) + |\varphi_i|$ total size: poly(n)

This is a log-space reduction.

The only part specific to x is the size of a config. and φ_0.

Also hard-coding START & ACCEPT configs.

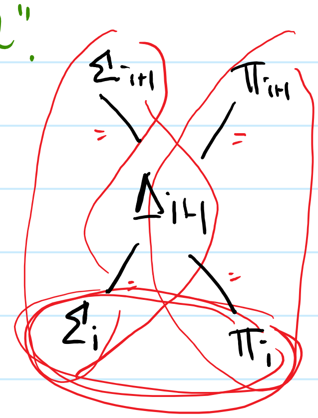
Final: $\exists z\ \forall x\ \forall y\ [(X=START \wedge Y=z) \vee (X=z \wedge Y=ACCEPT) \Rightarrow \varphi_{n^k}(X,Y)]$

Monday, May 14, 2018 7:49 AM

PH collapse:

Theorem: If $\Sigma_i = \Pi_i$ then for all $j > i$
$$\Sigma_j = \Pi_j = \Delta_j = \Sigma_i.$$

"The polynomial hierarchy collapses to the i^{th} level".



Proof: It's sufficient to show: $\Sigma_i = \Sigma_{i+1}$

$$L \in \Pi_{i+1} \Rightarrow \bar{L} \in \Sigma_{i+1} \Rightarrow \bar{L} \in \Sigma_i \Rightarrow L \in \Pi_i$$

Then: $\Sigma_{i+2} = NP^{\Sigma_{i+1}} = NP^{\Sigma_i} = \Sigma_{i+1}$, etc...

Now to show $\Sigma_i = \Sigma_{i+1}$

$L \in \Sigma_{i+1}$ iff L is expressible as:
$$L = \{x \mid \exists y \ R(x,y) \text{ accepts}\}$$
$$R \in \Pi_i$$

Hypothesis: $\Pi_i = \Sigma_i$

$$L(R) = \{(x,y) \mid \exists z \ (x,y,z) \in R'\} \quad R' \in \Pi_{i-1}$$

$$L = \{x \mid \exists y,z \ (x,y,z) \in R'\} \quad R' \in \Pi_{i-1}$$
$$\Rightarrow L \in \Sigma_i$$

In the PH, almost all "natural" complete problems lie in the 2^{nd} or 3^{rd} tier of the hierarchy.