CSE 431
Intro to Theory of Computation

So far

- $TIME(f(n)) \subseteq NTIME(f(n)) \subseteq SPACE(f(n)) \subseteq NSPACE(f(n)) \subseteq TIME(2^{O(f(n))})$ (above NTIME: coNTIME(f(n))) (for $f(n) \geq \log_2 n$)

- $NSPACE(f(n)) \subseteq SPACE(f^2(n))$ for $f(n) \geq \log_2 n$

In particular,

$$P \subseteq \begin{matrix} NP \\ coNP \end{matrix} \subseteq PSPACE \subseteq EXP$$

PSPACE = NSPACE

Def^n B is <u>PSPACE-hard</u> iff $\forall A \in PSPACE$, $A \leq_m^P B$

Def^n B is <u>PSPACE-complete</u> iff

- $B \in PSPACE$
- B is PSPACE-hard

Def^n $TQBF = \{\langle \Phi \rangle : \Phi$ is a fully quantified Boolean formula that evaluates to true$\}$

e.g. $\exists x_1 \forall x_2 \exists x_3 ((x_1 \to x_2) \wedge (x_2 \to x_3) \wedge (x_3 \to x_2))$

true, $x_1 = 0$, $x_3 = x_2$

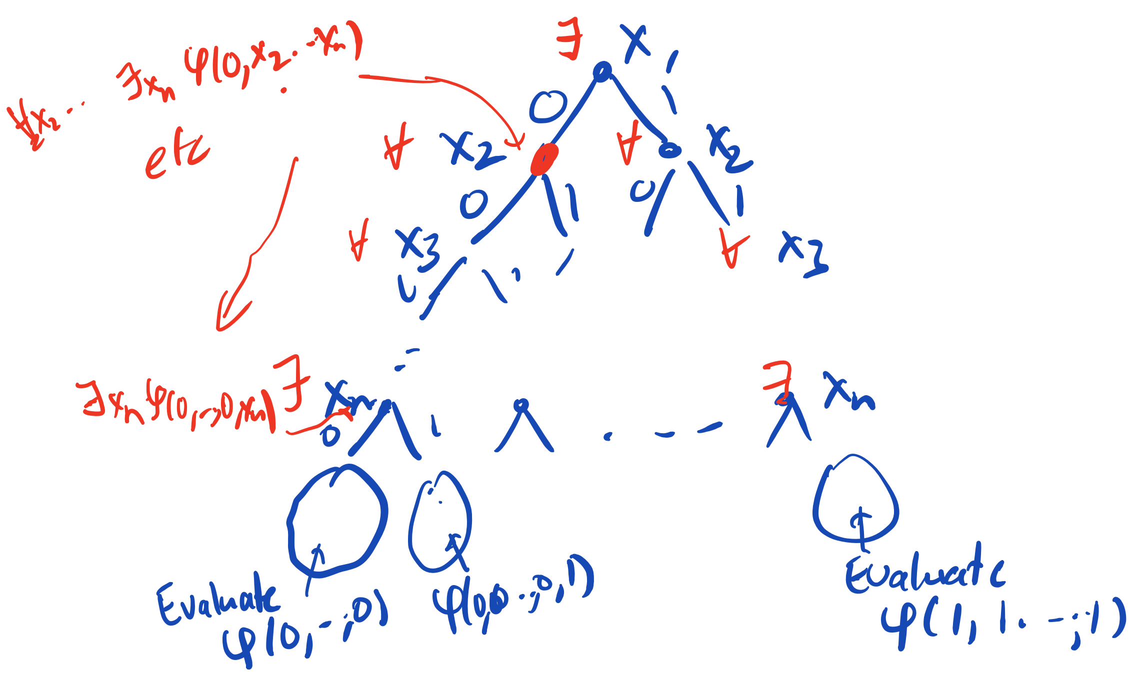
Thm TQBF is PSPACE-complete

Proof: Claim: TQBF ∈ PSPACE

Write $\Phi = Q_1 x_1 \cdots Q_n x_n \, \varphi(x_1 \cdots x_n)$

quantifier $Q_i = \exists$ or $\forall$ — Boolean formula on $x_1 \cdots x_n$

Imagine a full binary tree on the assignments to $x_1 \cdots x_n$



Consider an alg that does a (recursive) DFS on this tree evaluating the formula:

The value at the leaf is easy poly time to compute.
We can evaluate each node as we backtrack from the DFS

If x_i is labelled by $\exists$:
evaluate left-child
if left-child's value is 1 return 1
else evaluate right child and return its value

If x_i is labelled by $\forall$:
evaluate left-child
if left-child's value is 0 return 0
else evaluate right-child and returns its value

What storage is required:
DFS stack: height n
Enough to evaluate φ at a leaf $\langle \varphi \rangle$
Total $n + |\langle \varphi \rangle| \Rightarrow$ linear space

2) TQBF is PSPACE-hard:

Let $A \in PSPACE$
$\therefore A$ is decided by some TM M using space $S = cn^k$ for some constant c, k

Recall: $x \in A \Leftrightarrow \exists$ path from C_0 to C_{accept} in $G_{M,x}$
(Configuration Graph of M on input x)

- $G_{M,x}$ has at most $T = 2^{dS}$ nodes
- each node of $G_{M,x}$ is a configuration of M on input x and can be described by $O(S)$ bits. ($O(n^k)$)

Recall $\text{CANYIELD}_t(C,D)$ ← configuration of M on input x

$\equiv$ there is a path from C to D in $G_{M,x}$ of length $\leq t$.

$\text{CANYIELD}_0(C,D) \equiv$ "$C=D$"

$\text{CANYIELD}_1(C,D) \equiv$ "$C=D$" or "$C \vdash_M D$"
"yields in one step"

$\text{CANYIELD}_t(C,D) \equiv \exists C_{mid}.(\text{CANYIELD}_{\lceil t/2 \rceil}(C, C_{mid}) \wedge \text{CANYIELD}_{\lfloor t/2 \rfloor}(C_{mid}, D))$

We prove $A \leq_m^P$ TQBF

<u>Goal:</u> $x \in A \xrightarrow{f} \langle \Phi_{M,x} \rangle$

where $\Phi_{M,x} \equiv 1$ iff $\text{CANYIELD}_T(C_0, C_{accept})$

We will define formulas $\Phi_t(\vec{C}, \vec{D})$ s.t.

$\Phi_t(\vec{C}, \vec{D})$ iff $\text{CANYIELD}_t(C,D)$

where $\vec{C}, \vec{D}$ are binary vectors of variables corresponding to configs C, D

Since space is $\leq S$, $\vec{C}, \vec{D}$ take $O(S) = O(n^k)$ bits.

We will set $\Phi_{M,x} = \Phi_T(C_0, C_{accept})$ ← constant bit-vectors each representing specific configuration.

$\Phi_0(\vec{C},\vec{D})$ is an $\wedge$ of $O(S)$ conditions of the form $(\vec{C})_i = (\vec{D})_i$

$\Phi_1(\vec{C},\vec{D}) = \Phi_0(\vec{C},\vec{D}) \cup$ "$C \vdash_M D$"

easy to express in logic using δ function

(just like adjacent rows of in Cook-Levin tableau)

Assume wlog that we only define Φ_t when t is a power of 2.

Obvious attempt based on CANYIELD$_t(C, D)$

~~$\Phi_t(\vec{C},\vec{D}) = \exists \vec{C}_{mid} \left(\Phi_{t/2}(\vec{C},\vec{C}_{mid}) \wedge \Phi_{t/2}(\vec{C}_{mid},\vec{D}) \right)$~~

$O(S)$ $\exists$ quantifiers in a row for the bits of C_{mid}

When we unwind this recursion we realize that Φ_t

$\text{size}(\Phi_t) > 2\,\text{size}(\Phi_{t/2})$

So $\text{size}(\Phi_t) > t$ which will be bad for Φ_T since T is exponential and we need to be computable in polytime

But we haven't used any $\forall$ in this!

Our new idea will be to write $\Phi_{t/2}$ just once and use the $\forall$ quantifier to cover the two cases:

Define $\Phi_t(\vec{C},\vec{D}) = \exists \overrightarrow{C_{mid}}.\ \forall \vec{E},\vec{F}$

$$\Big[\Big(\big((\vec{E}=\vec{C}) \wedge (\vec{F}=\overrightarrow{C_{mid}})\big) \vee \big((\vec{E}=\overrightarrow{C_{mid}}) \wedge (\vec{F}=\vec{D})\big)\Big) \rightarrow \Phi_{t/2}(\vec{E},\vec{F})\Big]$$

the two cases we care about

Now $size(\Phi_t) = cn^k + size(\Phi_{t/2})$

$\therefore size(\Phi_t) = (cn^k) \cdot \underbrace{\log T}_{O(n^k)} + cn^k$

$\therefore size(\Phi_t)$ is $O(n^{2k})$ which is polynomial

Φ_t is very easy to write down
- everything but Φ_1 doesn't even depend on the details of M

$\therefore f$ is polynomial

By construction it satisfies correctness ∎

Next time: complexity classes inside P.
- is every problem in P solvable in small space?