Computational Complexity – Lecture 10.

Agenda:
- More about the polynomial hierarchy
  - Defn via oracles
  - Alternating TMs.
- Time-space tradeoffs for SAT.

Recap:
- $\Sigma_2^P = \{ L : x \in L \Leftrightarrow \exists y . \forall z \ M(x,y,z) = 1 \}$.

  $\Pi_2^P = \{ L : x \in L \Leftrightarrow \forall y \ \exists z \ M(x,y,z) = 1 \}$

  $|||^{rly}$ for other Σ_i^P, Π_i^P.

- $\Sigma_i = \Pi_i \Rightarrow$ PH collapses to Σ_i / Π_i.

  Does not extend to PSPACE $= \Sigma_i$!

Σ_2^P and oracles:

Qn: Is $\Sigma_2^P = NP^{NP}$?

Attempt towards showing this:

$\subseteq$: $L \in \Sigma_2^P$:

$$x \in L \Leftrightarrow \exists y . \forall z \ M(x,y,z)$$

NP^{SAT}: Guess y.

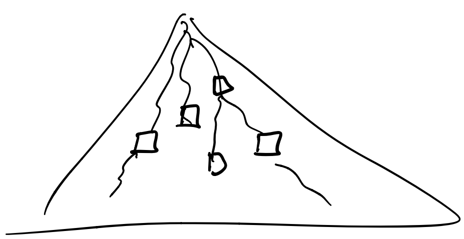
Ask oracle $\exists z . M(x,y,z) = 0$

If oracles says no, you accept.

$\supseteq$: Main issue: NP^{NP} can make many queries...
But Σ_2^p feels like a "single query"

Idea: Use the nondeterminism to guess answers when possible.

Let N be an NP^{SAT} machine accepting L.



- Guess a non det path of N
- Guess answers to all queries.
- Guess assignments for "yes" answers.
- Confirm "no" answers.

$$\underbrace{\exists x_1, \ldots, x_m}_{\text{non-det choices of } N.} \quad \underbrace{\exists a_1, \ldots, a_m}_{\text{answers to oracle queries}} \quad \underbrace{\exists r_1, \ldots, r_m}_{\text{assignments to "yes" answers}} \quad \underbrace{\forall s_1, \ldots, s_m}_{\text{for the "TAUT" query}}$$

If we run N on the non-det path $x_1, \ldots, x_m$ and if oracle answers were $a_1, \ldots, a_m$, does N accept x

and: for $i = 1 \ldots m$

$a_i = 1$ then r_i is a sat-ass. for the i^{th} query.

$a_i = 0$ then s_i is a falsifying assignment for i^{th} query.

$\Rightarrow L \in \Sigma_2^p$. $\square$

Thm: $\Sigma_2^P = NP^{NP} = NP^{\Sigma_1^P}$

For $i \geq 2$ $\quad \Sigma_i^P = NP^{\Sigma_{i-1}^P}$

Which language? What is a complete problem here?

Σ_2-SAT $= \{ \text{" } \exists x \; \forall y \;\; \varphi(x,y) \text{ "} : \text{ that are true} \}$

Obs: This is complete for Σ_2^P.

Qn: What is a complete problem for PH?

Say $L \in PH$ is a complete problem. $\Rightarrow$ $L \in PH = \bigcup \Sigma_i^P$

Any $L' \in PH$ $\quad L' \leq_p L$

$\Rightarrow$ there is some i: $L \in \Sigma_i$

But Σ_{i+1}-SAT $\in \Sigma_{i+1} \subseteq PH$

$\Rightarrow$ Σ_{i+1}-SAT $\leq_p L$ $\Rightarrow$ $\Sigma_{i+1}^P = \Sigma_i^P$

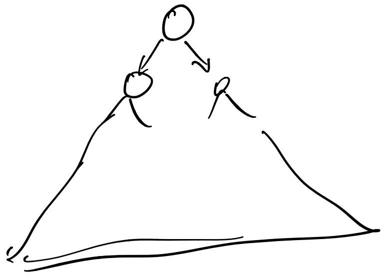
$\Rightarrow$ PH collapses.

If you believe PH does not collapse, then there are no complete problems.

Alternating TMs: (combining non-determinism & co-nondeterminism)

Non-det TM: Accepts an input if some path leads to accept.

Co-nondet TM: Accepts an input if every path leads to accept.



Machine state will determine if it is in a $\exists$ mode or $\forall$ mode.

Alternating TMs: Like regular TMs with potentially two transitions from each configuration. Every state except accept, reject is labelled with either $\exists$ or $\forall$.

An ATM accepts an input if ~~~~

Eg: ATM for SAT.

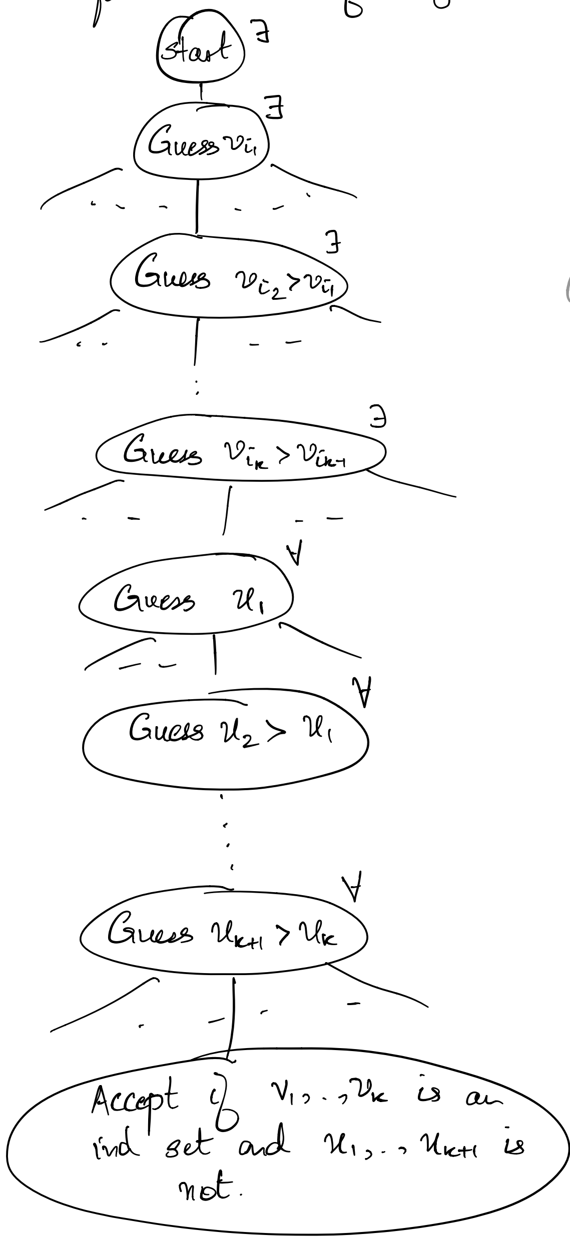
ATM for TAUT

Replace all states with $\forall$

A configuration C is accepting if

- C is in q_{accept}.
- If state labelled by $\exists$, then at least one of its children is accepting.
- If state is labelled by $\forall$, then <u>all</u> its children are accepting.

Eg: ATM for ind. set of size exactly k.



Can also write something similar for Σ_i^P.

$\mathrm{ATIME}(t(n)) = \{ L :$ L accepted by an ATM running in time $O(t(n))$ in every comp. path $\}$

$\mathrm{ASPACE}(s(n)) = \{ L :$ - - - using space $O(s(n))$ in every comp path. $\}$

$\Sigma_i\text{-Time}(t(n)) = \{ L :$ acc. by ATM in time $O(t(n))$ but with atmost i alternations, & starts with Σ $\}$.

$\Sigma_i^P = \bigcup_{c \geq 0} \Sigma_i\text{-TIME}(n^c)$

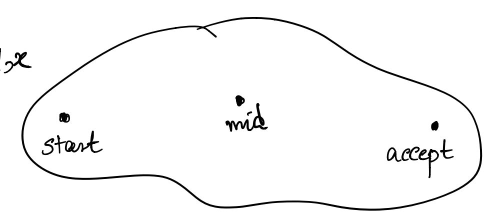
What can you say about ATIME(poly(n)) ? PSPACE! (TQBF)

Thm: $\text{NSPACE}(s(n)) \subseteq \text{ATIME}(s(n)^2) \subseteq \text{SPACE}(s(n)^2)$

Pf: $\subseteq$: Just do a DFS on the computation graph.

$\subseteq$: Similar to Savitch's theorem.

$G_{M,x}$



Reach(start, accept, T) $\quad T = 2^{O(s(n))}$

- ▷ Guess ($\exists$) mid.
- ▷ Guess ($\forall$) $\{$(start, mid, $T/2$), (mid, accept, $T/2$)$\}$

Verify Reach$(u, v, T/2)$

$$AT(s, T) = O(s) + AT(s, T/2)$$
$$= O(s \log T) = O(s^2).$$

$\square$

The connection goes the otherway too.

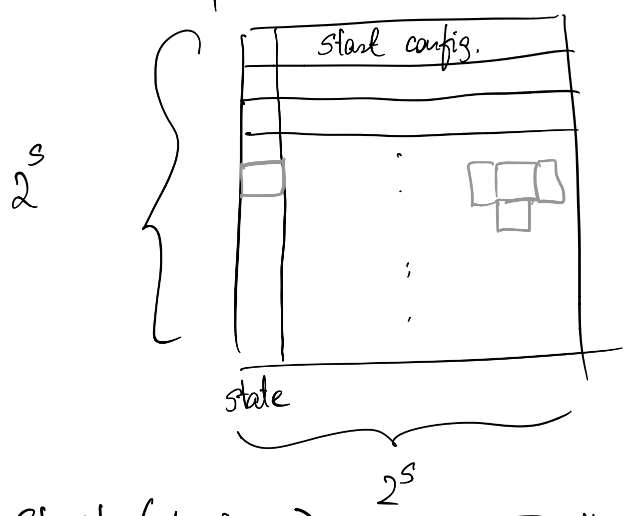
Thm: $\text{ASPACE}(s(n)) = \text{TIME}(2^{O(s(n))})$ if $s(n)$-space constr.

Pf: $\subseteq$: Exactly like $\text{space}(s(n)) \subseteq \text{TIME}(2^{O(s(n))})$.

This is easy.

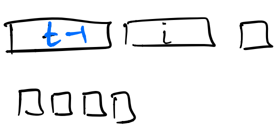
2: $\mathrm{TIME}(2^s) \subseteq \mathrm{ASPACE}(O(s))$

Computational Tableau:



Check (t, i, σ) : Is the i^{th} cell in the t^{th} row σ?

- Guess $(\exists)$ $q, \sigma_1, \sigma_2, \sigma_3$ — $O(1)$
- Det. verify if $(q, \sigma_1, \sigma_2, \sigma_3) \longmapsto \sigma$ — $O(1)$
- Forall:



Check $(t-1, 1, q)$ — $\log t = s$ bits.
Check $(t-1, i-1, \sigma_1)$
Check $(t-1, i, \sigma_2)$
Check $(t-1, i+1, \sigma_3)$

How much space are we using? $O(s)$.

$\therefore \mathrm{TIME}(2^s) \subseteq \mathrm{ASPACE}(O(s))$. $\square$.

Given classes like ATIME, ASPACE etc, is there a hierarchy thm here?

Yes: If $f(n+1) = o(g(n))$, time constr. then
$$\Sigma_k\text{-TIME}(f) \subsetneq \Sigma_k\text{-TIME}(g).$$
Pf exactly the same.

An easier thing to prove:

$\forall k \geq 1$. $$\Sigma_k\text{-TIME}(f(n)) \not\subseteq \Pi_k\text{TIME}(o(f(n))).$$
"No complementary speed-up".

An application: Time-space tradeoffs.

Is $NP = L$? Probably no.
Does SAT take superpolynomial time? Probably yes.

Suppose I force the TM to only use $\log^2 n$ space.
Can we prove time lower bounds here?

$$\text{TISP}(t(n), s(n)) = \{L : \text{accepted by a TM that with time bound } t(n) \text{ \& space bound } s(n)\}$$
NOT $\text{TIME}(t(n)) \cap \text{SPACE}(s(n))$

Thm: $\text{NTIME}(n) \not\subseteq \text{TISP}(n^{1.4}, n^{0.01})$ $\longrightarrow$ Next class!

Current best of this type: $\text{NTIME}(n) \not\subseteq \text{TISP}(n^{1.8\ldots}, n^{o(1)})$.
[Williams].