Def: $\Sigma_i \mathrm{TIME}(T)$ ($\Pi_i \mathrm{TIME}(T)$) is the set of all languages L decided by a time T ATM M with initial state labeled $\exists$ (resp., $\forall$) such that M is i-alternating.

Lemma: $\forall i \in \mathbb{Z}^+$

$$\Sigma_i^P = \bigcup_c \Sigma_i \mathrm{TIME}(n^c)$$

$$\Pi_i^P = \bigcup_c \Pi_i \mathrm{TIME}(n^c)$$

Cor: $PH = \bigcup_i \bigcup_c \Sigma_i \mathrm{TIME}(n^c)$

Unlimited # of Alternations

- This is ATIME(T)

- Alternating Polynomial Time

$$AP = \bigcup_c ATIME(n^c)$$

Theorem: AP = PSPACE

Proof. (1) PSPACE $\subseteq$ AP

- Show TQBF $\in$ AP

$\phi \in TQBF \leftrightarrow$

$$Q_1 x_1 \, Q_2 x_2 \cdots Q_m x_m \; \psi(\vec{x}) = 1$$

$Q_i \in \{\exists, \forall\} \; \forall i$

$\rightarrow$ Trivially in AP

↳ just guesses, check

everything else in poly-time

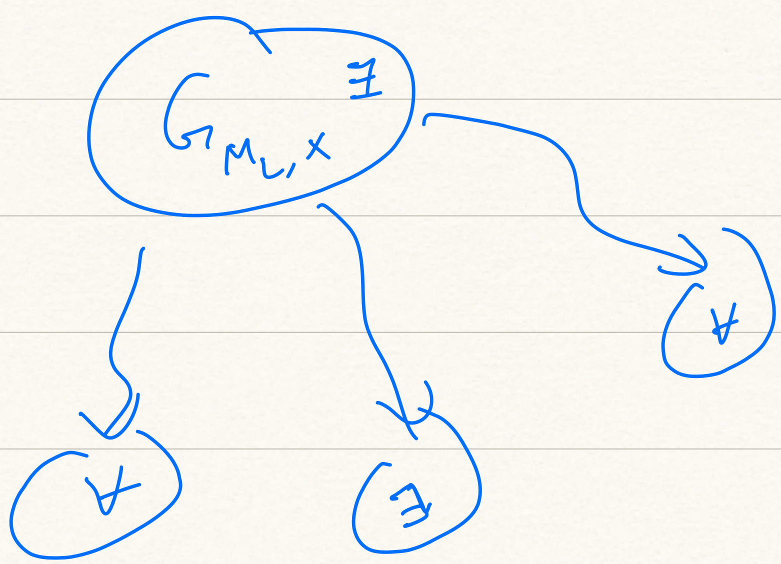
↳ Using the alternating states.

② $AP \subseteq PSPACE$

Let $L \in AP$.

- $\exists$ ATM M_L deciding L in poly-time
- Let D be a machine

$D(x)$:



- Traverse recursively

- DFS
→ Poly-space

$$\text{APSPACE} = \bigcup_c \text{ASPACE}(n^c)$$

Alternating poly-space

↳ unlimited # alternations

Theorem

APSPACE = EXP

AL = ASPACE(log(n))

same proof idea

Theorem AL = P

Time-Space Tradeoffs for SAT

- For SAT

  (1) Believe we need exp time (or at least super poly)

  (2) Believe we need linear space

- We can rule out an algorithm for solving SAT in linear time and log space

Thm: Let $S, T : \mathbb{N} \to \mathbb{N}$ be functions. Define $TISP(T,S)$

to be all languages decidable by a TM M using at most $O(T)$ time and $O(S)$ space.

Then, $SAT \notin TISP(n^{1.2}, n^{0.2})$

more generally

$SAT \notin TISP(n^c, n^d)$ for $c, d > 0$ and $c(c+d) < 2$.

Proof:

First, a claim.

Claim 1: $TISP(n^{12}, n^{2}) \subseteq \Sigma_2 TIME(n^8)$

- Proof similar to Savitch's thm + TQBF is PSPACE complete

- Let $L \in TISP(n^{12}, n^2)$ w/ TM $\underline{M_L}$
  - ↳ $M_L(x)$ runs in time $|x|^{12}$ and space $|x|^2$
- construct config graph $\underline{G_{M_L,x}}$
  - ↳ all configs need at most $|x|^2$ bits to describe
  - ↳ $M_L(x)=1 \iff \exists$ path in $G_{M_L,x}$ from $\underline{C_{start}}$ to some $\underline{C_{accept}}$
    - ↳ length $\leq |x|^{12}$

n^{12}

This path exists iff

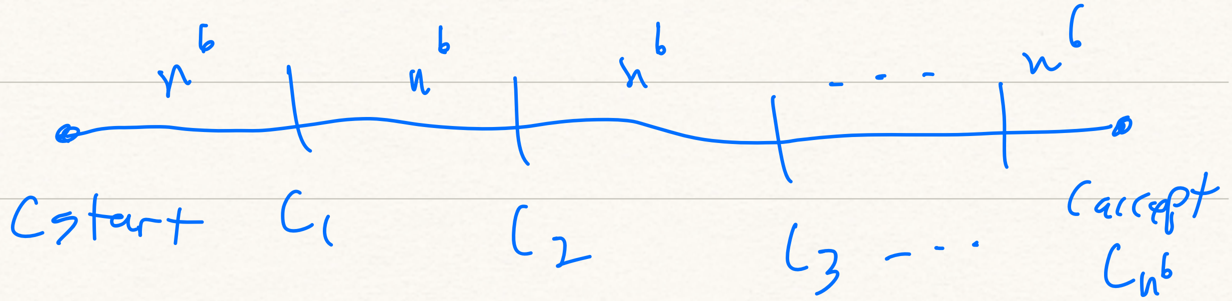
there exist n^6 configs.

$\underbrace{C_1, C_2, \dots, C_{n^6}}_{n^6 \cdot n^2 \text{ bits} = O(n^8)}$ such that

- if $C_0 = C_{start}$ then C_{n^6} is accepting
- and $\forall i \in [n^6]$

  C_i follows from C_{i-1} in at most n^6 steps

n^{12}



n^6 chunks of length

n^6 each

So $L \in \Sigma_2 TIME(n^8)$.

Claim ②: Suppose that

$$NTIME(n) \subseteq DTIME(n^{1.2}).$$

Then:

$$\Sigma_2 TIME(n^8) \subseteq NTIME(n^{9.6}).$$

Proof: $L \in \Sigma_2 TIME(n^8)$

$\iff \exists$ DTM M_L s.t.

$x \in L \iff \exists w_1 \in \{0,1\}^{c|x|^8}$

$\forall w_2 \in \{0,1\}^{d|x|^8}$

$M_L(x, w_1, w_2) = 1$

in time

$O(|x|^8)$

- By assumption,

  $NTIME(n) \subseteq DTIME(n^{1.2})$

- By a padding argument we have a DTM D that

  on input x, w_1 for
  $n = |x|$ $|w_1| = c$.
  $D(x, w_1)$ runs in time

  $$O((n^8)^{1.2}) = O(n^{9.6})$$

  and returns 1 iff
  $\exists\, w_2$ s.t $M_L(x, w_1, w_2) = 0$.

$\Rightarrow L \in NTIME(n^{9.6})$.

The goal has been to show

$$\mathrm{NTIME}(n) \not\subseteq \mathrm{TISP}(n^{1.2}, n^{0.2})$$

SAT

Suppose <u>not</u>. Then we get a contradiction.

Claim 1 + 2 give us

$$\mathrm{NTIME}(n^{10}) \subseteq \mathrm{TISP}(n^{12}, n^{2})$$
$$\subseteq \Sigma_2\mathrm{TIME}(n^{8})$$
$$\subseteq \mathrm{NTIME}(n^{9.6})$$

violates time hierarchy Non-det. $\square$

PH via Oracle Machines

thm: $\forall i \in \mathbb{N}$

$$\Sigma_i^p = NP^{\Sigma_{i-1} SAT}$$

$$\Sigma_{i-1} SAT = \{ \phi \mid \phi = \exists x_1 \forall x_2 \cdots Q_{i-1} x_{i-1}\, \psi(x_1, \dots, x_{i-1}) = 1 \}$$

Proof: For $i = 2$

$$\Sigma_2^p = NP^{SAT}$$

① let $L \in \Sigma_{2}^p$ (i). $\exists$ DTM M_L s.t.

Σ_{i-1}

$x \in L \iff \exists w_1 \boxed{\forall w_2 \; M_2(x, w_1, w_2) = 1}$

$\underbrace{\quad}_{\text{poly}(|x|)}$

- Fix x, w_1. Then

$\forall w_2 \; M_2(x, w_1, w_2) = 1$ is

a coNP statement.

→ negate it:

$\exists w_2 \; \neg M_2(x, w_1, w_2) = 1$

is NP statement

→ We have SAT oracle

→ check w/ 1 call

$\Rightarrow \Sigma_2^P \subseteq NP^{SAT}$

- Now, suppose $L \in NP^{SAT}$ w/ NTM N.
- $N(x)$ makes at most $\text{poly}(|x|)$ queries $\underbrace{q_1, \ldots, q_k}_{\text{SAT Formulas}}$ w/ answers $a_1, \ldots, a_k$.
  - * q_i can depend on all (q_j, a_j) $j \le i-1$ and any choices made by NTM so far
- Let $c_1, \ldots, c_m \in \{0,1\}$ denote non-det. choices made by $N(x)$.

- $x \in L \longleftrightarrow$

  $\exists$ choices $c_1, \ldots, c_m$ and answers $a_1, \ldots, a_k$ such that

  - If $N(x)$ makes choices $c_1, \ldots, c_m$ and receives oracle answers $a_1, \ldots, a_k$
  - Then ① $N(x)$ reaches an accepting state AND

    ② - If $a_i = 1$ then $\exists$ assignment u_i s.t. $q_i(u_i) = 1$

    - If $a_i = 0$ then $\forall$ assignments v_i $q_i(v_i) = 0$

where q_i is i^{th} query

$x \in L \iff$

$\exists c_1, \ldots, c_m, a_1, \ldots, a_k, u_1, \ldots, u_k$

$\forall v_1, \ldots, v_k$:

- N accepts x using choices $c_1, \ldots, c_m$ and
- $\forall i \in [k]$ if $a_i = 1$ then $q_i(u_i) = 1$

AND

- $\forall i \in [k]$ if $a_i = 0$ then $q_i(v_i) = 0$

$\Rightarrow L \in \Sigma_2^P$ $\square$

Lecture 13