Computational Complexity: Lecture 11.

Recap: Σ_i^P, Π_i^P

- defn via "quantified verifier"
- defn via oracle TMs.
- defn via Alternating TMs.

- Alternating TMs:
  - ▷ Thm: $NSPACE(s(n)) \subseteq ATIME(s(n)^2) \subseteq DSPACE(s(n)^2)$
  - ▷ Thm: $ASPACE(s(n)) = DTIME(2^{O(s(n))})$.
- Hierarchy theorems:

  TC $f(n+1) = o(g(n)) \Rightarrow \Sigma_i\text{-TIME}(f(n)) \subsetneq \Sigma_i\text{-TIME}(g(n))$.
- "No complementary speed-up"

  $\Sigma_i\text{-TIME}(f(n)) \not\subseteq \Pi_i\text{-TIME}(o(f(n)))$

  for any $i \geq 1$ and T.C $f(n)$.

Agenda:
- Time-space trade offs for SAT (NTIME(n)).
- Introduction to Boolean circuits.

Some annoying open questions

- ▷ Is SAT $\in$ L? We believe "no"
- ▷ Does SAT have an $O(n)$ time algorithm? We believe "No".

[Fortnow] We aren't wrong on both.

Thm: If SAT $\in$ L, then there is an $\varepsilon > 0$ s.t SAT $\notin$ TIME$(n^{1+\varepsilon})$.

A different kind of trade off:

Qn: Is SAT $\in$ TISP $(n, n^{o(1)})$.
$\hookrightarrow$ time $\hookrightarrow$ space?

[Fortnow, Fortnow-Lipton-van Melkebeek-Viglas, Williams, Diehl-van Melkebeek].

Thm: [Williams] NTIME $(n) \not\subseteq$ TISP $(n^{1.8\ldots}, n^{o(1)})$.

We won't quite get there but we will try and get close to it.

Thm: NTIME $(n) \not\subseteq$ TISP $(n^{c}, n^{o(1)})$

(we'll try and keep improving c).

key ingredients:

- "No complementary speed-up".

  Σ_k-TIME $(f(n)) \not\subseteq \Pi_k$-TIME $(o(f(n)))$

- "Alternation elimination"

- "Alternation trading".

Lemma: [Alternation elimination]. If NTIME $(n) \subseteq$ TIME (n^c) then Π_2 - TIME $(n) \subseteq \Pi_1$ TIME (n^c).

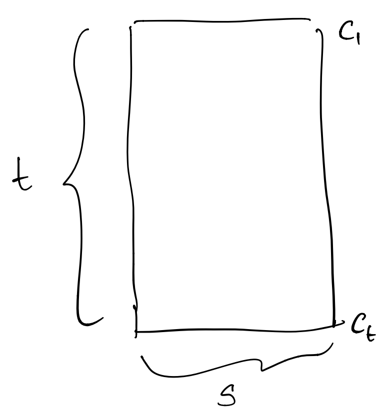
Pf: $\forall \ldots \forall \underbrace{\exists \ldots \exists \text{ deterministic}}_{\Sigma_1\text{-TIME}(n)} \rightsquigarrow \forall \ldots \forall$ Time (n^c)

Π_1- TIME (n^c). $\square$.

Lemma: (Alternation trading)

$$\mathrm{TISP}(t(n), s(n)) \subseteq \underset{\text{or } \Pi_2}{\Sigma_2}\text{-}\mathrm{TIME}\left(\sqrt{t(n)s(n)}\right)$$

Pf:



- $\exists\ C_r, C_{2r}, \ldots, C_{\frac{t}{r}\cdot r}$
- $\forall\ j \in \{1, 2, \ldots, t/r\}$
- $C_{(j-1)r} \rightsquigarrow C_{j,r}$ in r steps.

Total time: $O\left(\frac{t}{r} \cdot s\right) + O(\log(t/r))$
$+ \ O(r)$.

Optimised at $r = \sqrt{ts}$ to give total time $O(\sqrt{ts})$. $\square$.

Let's now prove the time space trade off:

AFSOC $\quad \underset{\Sigma_1\text{-TIME}(n)}{\overset{}{}}$ $\mathrm{NTIME}(n) \subseteq \mathrm{TISP}(n^c, n^{o(1)})$

(where $\Sigma_1\text{-TIME}(n) = \mathrm{NTIME}(n)$)

$$\Pi_2\text{-TIME}(n) \underset{\text{Alt, elim}}{\subseteq} \Pi_1\text{-TIME}(n^c)$$

$$\cap \text{ (Assumption + padding)}$$

$$\Sigma_2\text{-TIME}(n^{c^2/2 + o(1)}) \supseteq \mathrm{TISP}(n^{c^2}, n^{o(1)})$$

And will yield a contradiction if $c^2 < 1$ ie $c < \sqrt{2}$.

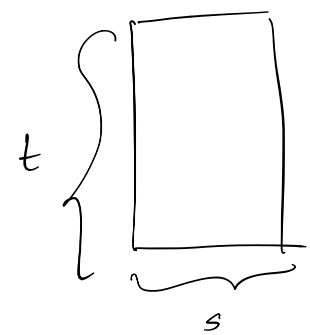
$\therefore$ $\mathrm{NTIME}(n) \not\subseteq \mathrm{TISP}(n^c, n^{o(1)})$ for any $c < \sqrt{2}$. $\square$.

Suppose $c > \sqrt{2}$, what can we say?

"New lemma": $\Pi_2\text{-TIME}(n) \subseteq \text{TISP}(n^{c^2}, n^{o(1)})$

"Better alt. elimination": $\Sigma_3\text{-TIME}(n) \subseteq \Sigma\text{-TISP}(n^{c^2}, n^{o(1)})$
$\subseteq \Sigma\Sigma\Pi\text{-TIME}(n^{c^2/2 + o(1)})$

Revisiting alternation trading:



$\exists\ C_r, C_{2r}, \ldots, C_{t/r \cdot r}$

$\forall\ j \in [t/r]$.

$C_{(j-1)r} \overset{r}{\rightsquigarrow} C_{jr}$ } TISP(r, s)

$\therefore\ \exists\ \forall\ \underbrace{\forall\ \exists\text{-det}}_{\Pi_2\text{-TIME}(\sqrt{rs})}$

Overall, $O\left(\frac{ts}{r} + \sqrt{rs}\right) \longrightarrow O(t^{1/3} s^{2/3})$

$\therefore\ \text{TISP}(t, s) \subseteq \Sigma_3\text{-TIME}(t^{1/3} s^{2/3})$

$\subseteq \Sigma_k\text{-TIME}(t^{1/k} s^{k-1/k})$.

Back to time-space tradeoffs.

AFSOC $\exists_1\text{-TIME}(n) \subseteq \text{TISP}(n^c, n^{o(1)})$

$\Sigma_3\text{-TIME}(n) \subseteq \Sigma_2\text{-TIME}(n^{c^2/2 + o(1)})$

$\subseteq \text{TISP}(n^{c^2/2 \cdot c^2 + o(1)}, n^{o(1)})$

$\subseteq \Pi_3\text{-TIME}(n^{c^4/6 + o(1)})$.

$\therefore$ We get a contradiction if $c^4 < 6$

ie $c < \sqrt[4]{6} \approx 1.565\ldots$

$\hookrightarrow 2^{1/4} \cdot 3^{1/4}$.

If $c \geq \sqrt[4]{6}$, we get a "new facts".

$$\Sigma_3\text{-TIME}(n) \subseteq \mathrm{TISP}(n^{c^4/2 + o(1)}, n^{o(1)})$$

$$\Sigma_4\text{-TIME}(n) \subseteq \Sigma_3\text{-TIME}(n^{c^4/6 + o(1)})$$

$$\begin{aligned}\therefore \Sigma_4\text{-TIME}(n) &\subseteq \Sigma_3\text{-TIME}(n^{c^4/6 + o(1)}) \\ &\subseteq \mathrm{TISP}(n^{c^4/6 \cdot c^4/2 + o(1)}, n^{o(1)}) \\ &\subseteq \Pi_4\text{-TIME}(n^{c^8/12 \times 4 + o(1)})\end{aligned}$$

We get a contradiction if $c^8 < 48$ or $c \approx 1.62$

or we get "new facts" ... and so on!

Eventually limits to $2^{1/4}\, 3^{1/8}\, 4^{1/16}\, 5^{1/32} \ldots \approx 1.6617\ldots$

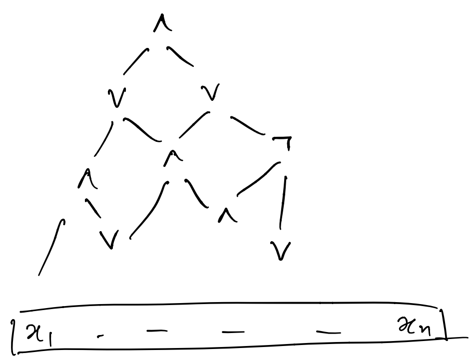
[William] : What is the best constant that can be obtained via this "assumption + alt. elimination + alt-trading + no comp. speedup"

Showed $\mathrm{NTIME}(n) \not\subseteq \mathrm{TISP}(n^c, n^{o(1)})$

for all $c < 2\cos(\pi/7)$. $\approx 1.8\ldots$

New topic: Boolean circuits

What is a circuit?



A DAG made up of $\wedge, \vee, \neg$ with leaves labelled $x_1, \ldots, x_n$.
Output computes $f : \{0,1\}^n \to \{0,1\}$.

Qn: Does this boolean circuit solve, say, SAT?

Hunh? We only have length n for input!

Defn: (Circuit family) $\mathcal{C} = \{C_i\}_{i=1 \ldots n}$ is a circuit family if C_i has i-inputs.
We say $\mathcal{C}$ is a family of size $S(n)$ if
$|C_i| \leq S(i) \quad \forall i$.
We say $\mathcal{C}$ "computes" $f : \{0,1\}^* \to \{0,1\}$ if

$$\forall x \in \{0,1\}^* \quad |x| = i \Rightarrow f(x) = C_i(x).$$

Fact: Every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by circuits of size $O(n \cdot 2^n)$. (In fact $O(2^n/n)$ is enough.)

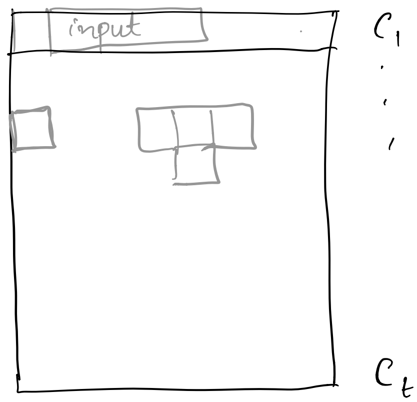
P/poly = class of languages that can be decided by a poly-size circuit family.

$$= \bigcup_{c \geq 0} \underline{\mathrm{SIZE}(n^c)}$$

languages dec. by ...

Thm: $P \subseteq P/poly$.

Pf: Very similar to Cook-Levin.



Each local computation can be "encoded" by a constant sized circuit.

Composing all gives a circuit of size $O(T(n)^2)$.

$\square$.

Claim: L is any unary language ie $L \subseteq \{1^n : n \in \mathbb{N}\}$.

Then, $L \in P/poly$.

Pf: $C_i = \begin{cases} \text{AND}(x_1, \ldots, x_i) & \text{if } 1^i \in L \\ 0 & \text{if } 1^i \notin L \end{cases}$

is clearly a circuit family deciding this language.

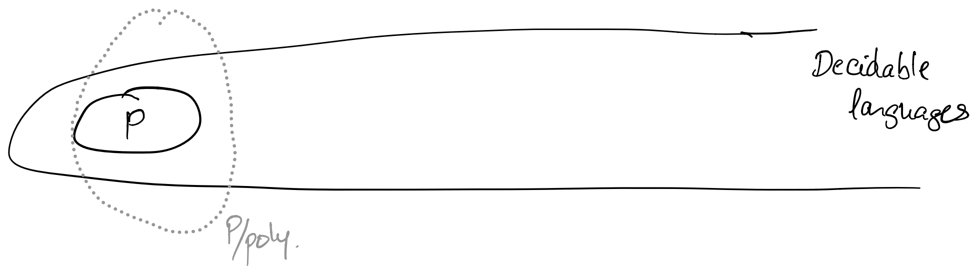
$\square$.

Cor: $P/poly$ contains languages that are undecidable!

Pf: $\text{UHALT} = \{ 1^n : \text{the } n^{th} \text{ machine halts} \}$.

is unary

is undecidable. $\square$



Qn: Is $NP \subseteq P/poly$? Probably no... but what if?

Thm: [Karp-Lipton] If $NP \subseteq P/poly$, then $PH = \Sigma_2$.

Next class:

- Pf of the Karp-Lipton theorem + extensions
- More on circuits & TMs with advice.