PH continued

Thm: Σ_i-SAT is Σ_i^P-complete

Σ_i-SAT is the set of true formulas where

$$\exists u_1 \forall u_2 \cdots \phi(u_1, u_2, \ldots, u_i)$$

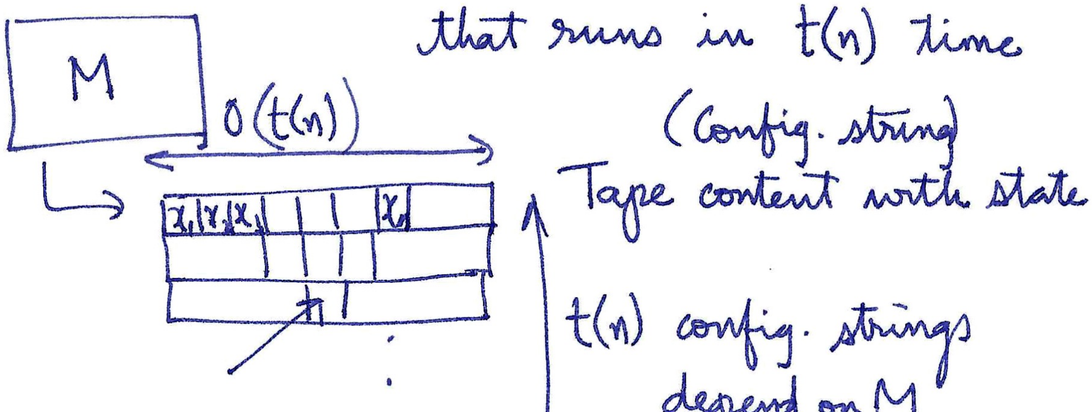
i quantifiers

set of variables

(tableau)

Proof: Let us go back to the Cook-Levin construction.

M is deterministic TM
that runs in $t(n)$ time



Each symbol in tableau is a fixed fn of a CONSTANT number of symbols/bits in previous config. string.

We can write a formula ϕ that encodes all these functions

M accepts $x \iff \phi(x)$ is true. $|\langle\phi\rangle| = O(t(n)^2)$

ϕ can be constructed in logspace. $(\log t(n))$

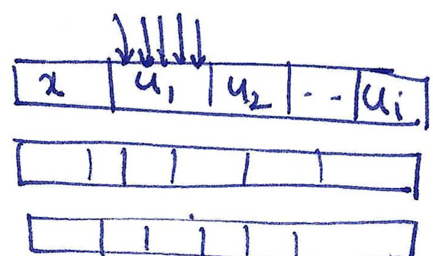
Given M and a ~~size~~ input size n and input x, there is an algorithm that constructs a formula $\phi_{M,x}$ ~~s.t.~~ in time poly$(t(n))$ and space $(\log(t(n)))$ s.t.

$$M \text{ accepts } x \iff \cancel{\phi_{M,n}(x) \text{ is true}}\ \exists\, y \text{ (setting of variables) } \phi_{M,x}(y) \text{ is true.}$$

Consider $L \in \Sigma_i^P$. There exists a polynomial p and a polytime TM M s.t.

$$x \in L \text{ iff } \underbrace{\exists u_1 \forall u_2 \ldots}_{i \text{ quantifiers}} (M(x, u_1, u_2 \ldots u_i) \text{ accepts})$$

Given M and x, we can construct a formula $\Phi_{M,x}$ s.t.



$$M(x, u_1, \ldots, u_i) \text{ accepts} \iff \Phi_{M,x} \text{ is satisfiable AND the first config. has } u_1, u_2 \ldots, u_i$$

x is in L iff

$$\exists u_1 \forall u_2 \ldots u_i\ M(x_1, u_1, \ldots, u_i) \text{ accepts}$$

$$\iff \exists u_1 \forall u_2 \ldots u_i \left(\Phi_{M,x} \text{ is satisfiable AND first config. has } u_1, u_2 \ldots u_i\right)$$

$$\exists u_1 \forall u_2 \ldots u_i\ \boxed{\Phi_{M,x}(u_1, u_2 \ldots, u_i) \text{ is satisfiable}}$$

$$\iff \exists u_1 \forall u_2 \ldots \exists u_i\, \exists T \text{ tableau s.t. } (\Phi(u_1, u_2 \ldots u_i, T) \text{ is true})$$

If i is odd: we get a Σ_i-SAT instance.

If i is even: $(\exists u_1 \forall u_2 \exists$ tableau - - $)$ we get Σ_{i+1}-SAT instance.

$\forall$ tableau, Σ_i-SAT instance

$\Phi_{M,x}(u_1, u_2 - , u_i)$ has a UNIQUE satisfying assignment (only one possible computational path)

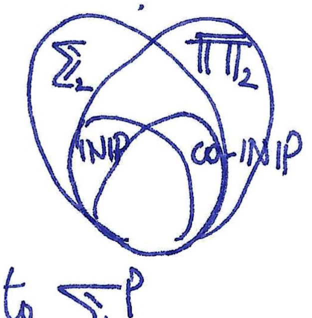
The variables in $\Phi_{M,x}$ are $\underbrace{u_1, u_2 - , u_i}_{u_1, - u_i \text{ from certificates}}$ & T ← tableau

For a given $u_1, u_2 - -, u_i$, $\exists$ unique T that forms a valid computational path

$\exists T (\Phi_{M,x}(u_1, - - u_i)$ is true$) \equiv \forall$ tableaus T [if tableau is valid then $\Phi_{M,x}(u_1 - -, u_i)(T)$ is true]

$\mathbb{P} \neq \mathbb{NP}$, $\mathbb{NP} \neq$ co-$\mathbb{NP}$

Generalization: (non-)collapse of $\mathbb{PH}$

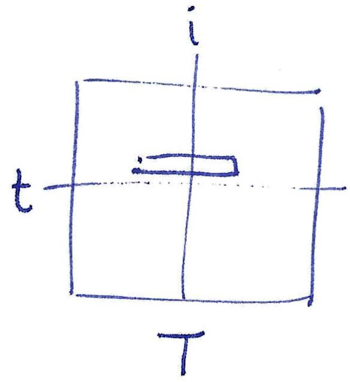


Non-collapse: All classes Σ_i^P, Π_i^P are distinct.

Clm: If $\Sigma_i^P = \Pi_i^P$, then $\mathbb{PH}$ collapses to Σ_i^P.

If M is deterministic, each symbol of tableau can be computed by a fixed function of a CONSTANT number of symbols from prev. config. string.

$$T_{t,i} = f\left(\left[T_{t-1,\, i-\Theta(1)}, \ldots, T_{t-1,\, i+\Theta(1)}\right]\right)$$

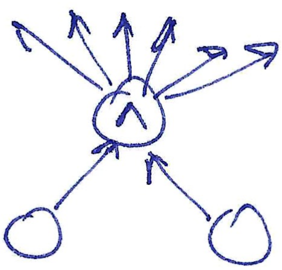


computed by a circuit of $O(1)$ size

Circuit: DAG, where leaves are labeled with inputs and intermediate nodes are labeled with AND, OR, NOT gates

The "fan-in" is the # inputs to each gate = 1 or 2

The "fan-out" can be unbounded.



All entries in T can be computed by a polynomial sized circuit. (So can the "output", acc/rej.)

Thm: Given M and an input size n, there is an algorithm that constructs a $O((t(n))^2)$ sized circuit $C_{M,n}$ in $\mathrm{poly}(t(n))$ time and $O(\log t(n))$ space s.t.

$$M \text{ accepts } x \text{ iff } C_{M,n}(x) = 1$$

Def: A circuit C_n on n inputs computes $f:\{0,1\}^n \to \{0,1\}$ if $\forall$ x of length n, $C_n(x) = f(x)$

Def: A language L is ~~def~~ decided by a circuit family $\mathcal{C} = \{C_1, C_2 \ldots\}$ s.t. $\forall$ x of length n, $x \in L$ iff $C_n(x) = 1$

A circuit family $\mathcal{C}$ has size $s(n)$ if
$\forall n$, $\text{size}(C_n) = O(s(n))$

size = # gates/nodes

Def: SIZE$(s(n))$ is the class of languages decided by circuit families of size $s(n)$.

$$\mathbb{P}/\text{poly} = \bigcup_{c \in \mathbb{N}} \text{SIZE}(n^c)$$

$\boxed{\textbf{Thm}: \quad \mathbb{P} \subsetneq \mathbb{P}/\text{poly}}$

Proof: We showed above that $\mathbb{P} \subseteq \mathbb{P}/\text{poly}$.

Every UNARY language is in $\mathbb{P}/\text{poly}$.
(Why? If $1^k \in L$, C_k does an AND of all inputs.
$1^k \notin L$, C_k outputs 0.)

UNARY-HALT $= \{ 1^k \mid k = \langle M, x \rangle \in$ Halting language ~~problem~~ $\}$

is undecidable but in P/poly. $\blacksquare$

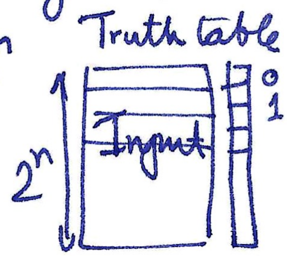
Circuits are more "convenient" from lower bound perspective.

<u>Thm [Shannon 49, Lyapunov]</u>: For every n, there exists a function f_n that requires a circuit of size $\Omega\left(\frac{2^n}{n}\right)$ to compute it.

Moreover, for ALL functions $f : \{0,1\}^n \to \{0,1\}$, f can be computed by a circuit of size $O\left(\frac{2^n}{n}\right)$.

<u>Proof</u>: (Part 1) Counting argument. There are more functions on n inputs than there are circuits with $2^n/n$ gates.

How many n input (Boolean) functions? 2^{2^n}



How many circuits with n inputs and $\cancel{2^n}$ ~~gates~~ $\le s$ gates?

Representation:

(1) Specify label (input variable) of each leaf/source

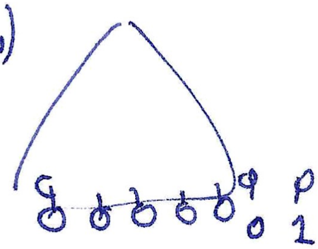
(2) Specify type of each gate/node

(3) Specify outneighbors of each gate/node

(in neighbors)

(1) Source is ~~tabled~~ labeled $x_1, \ldots, x_n, 0, 1$

$(n+2)^{(n+2)}$ (Hardcoding input sources)



(2) $\leq 3^s$

(3) $\leq s^{2s}$

(both)

s^2 choices of in-neighbors for each node

# of different circuits with s nodes

$$\leq \cancel{(n+2)^{(n+2)}} \times 3^s \times s^{2s} = 2^{\Theta(\cancel{\log s}\, s + s \log s)}$$

$$\leq 2^{5 s \log_2 s} = 2^{\Theta(s \log s)} \leq$$

Suppose $s = \dfrac{2^n}{cn}$

constant

$$\boxed{\begin{aligned} & 2^{5 s \log_2 s} \\ & = 2^{\left[5 \times \frac{2^n}{cn} \times \left(n - \cancel{\log_2(cn)}\right)\right]} \\ & < 2^{2^n} \end{aligned}}$$

Hence, there is a function (on n inputs) that is not computed by a circuit of size ($\leq$) $\dfrac{2^n}{cn}$. ∎