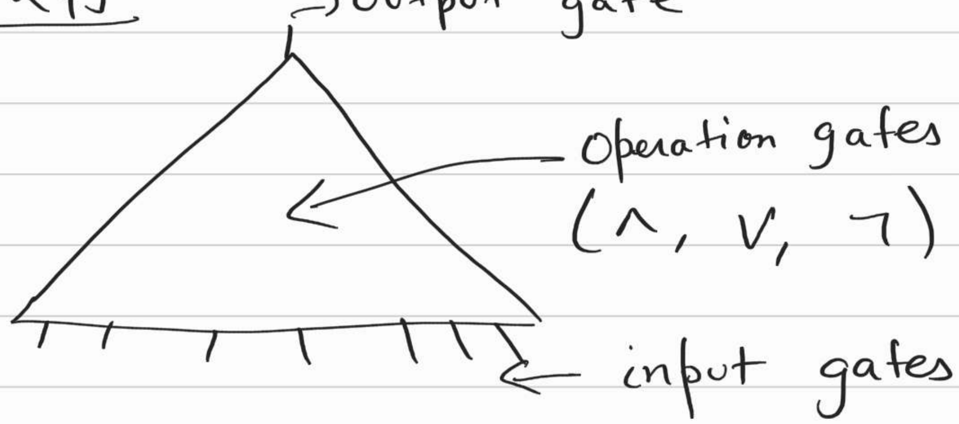
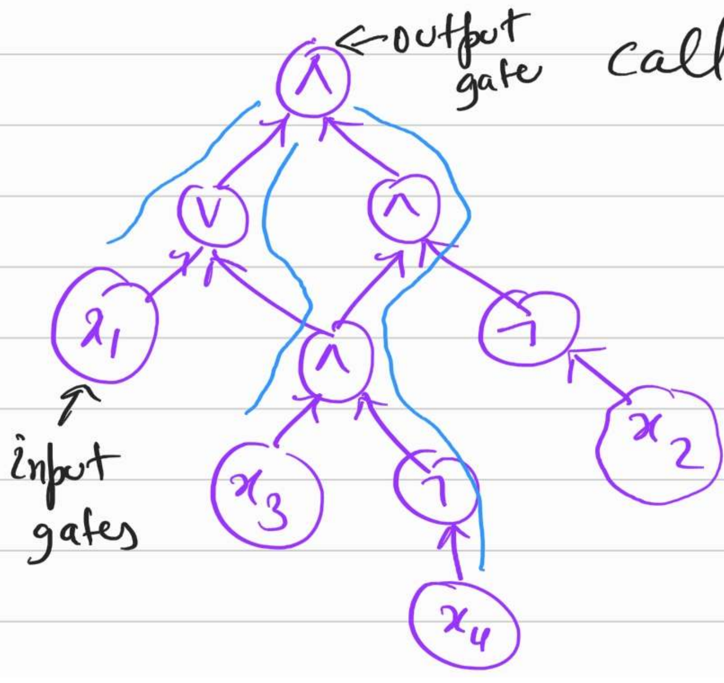
26/10/21 : Ckts



— Boolean Ckts.

- It is a directed acyclic graph (DAG).
- nodes with indegree 0 are called inputs. They are labelled with input variables.
- nodes of indegree 1 are operation gates labelled with NOT
- Nodes of indegree 2 are operation gates labelled with AND ($\wedge$) or OR($\vee$).
- Nodes with outdegree 0 are called output gates.



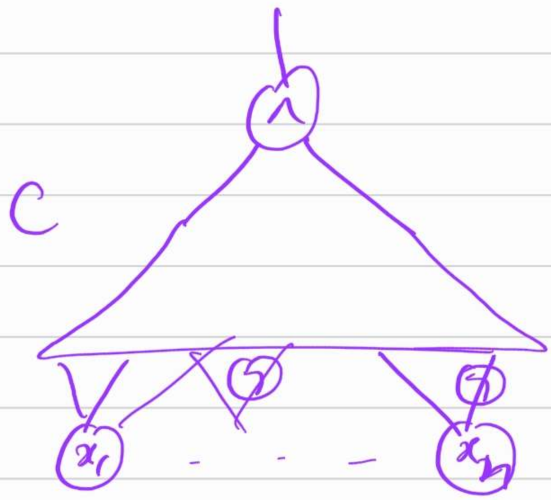
Complexity measures

(i) Size = # nodes

WLOG we will only count $\wedge$, $\vee$ gates.

(ii) Depth = length of the longest path from output node to any input gate.

(Not gates don't count towards depth).



$S \subseteq \{0,1\}^n$

where $S = \{x \in \{0,1\}^n \mid C(x) = 1\}$

A language $L \subseteq \{0,1\}^*$.

In other words a ckt has fixed input length.

<u>Circuit family</u> $:= \{C_0, C_1, C_2, \ldots, C_n, \ldots\}$

a language $L \subseteq \{0,1\}^*$ is recognized by a ckt family $\{C_n\}_{n \geq 0}$ if

$\forall n \geq 0, \ \forall x \in \{0,1\}^n \quad x \in L \Leftrightarrow C_n(x) = 1$

C_n exactly recognizes $L \cap \{0,1\}^n$.
(decides)
(computes)

On every input $y \in \{0,1\}^n$, C_n will either output 1 or 0.

$f : \{0,1\}^n \to \{0,1\}$

family of functions. $\{f_n\}_{n \geq 1}$

$f_n(x) = 1$ iff $x \in L$.

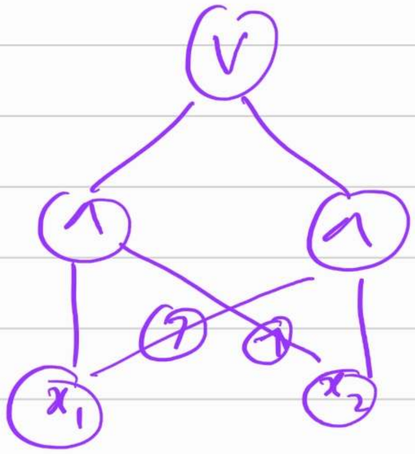
a family of ckts $\{C_n\}$ computes a family of functions $\{f_n\}$.

$$\text{PARITY}_n : \{0,1\}^n \longrightarrow \{0,1\}$$

$$\text{PARITY}_n(x_1, .., x_n) = \begin{cases} 1 & \text{if \# 1's in } x \text{ is odd} \\ 0 & \text{o/w} \end{cases}$$

$\{\text{PARITY}_n\}_{n \geq 1}$

$n=2$, $\quad \text{PARITY}_2 = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2)$



Defn:- We say that a language L is in $\text{SIZE}(s(n))$ if $\exists$ a $s(n)$-sized ckt family $\{C_n\}$ s.t. $\forall\, x \in \{0,1\}^n$

$$x \in L \iff C_n(x) = 1.$$

($s(n)$-sized ckt family means $\forall\, n \in \mathbb{N}$ $|C_n| \le s(n)$).

$\{\text{AND}_n\}$ — $\in \text{SIZE}(n)$.

Every function $f: \{0,1\}^n \to \{0,1\}$ can be computed by a ckt.

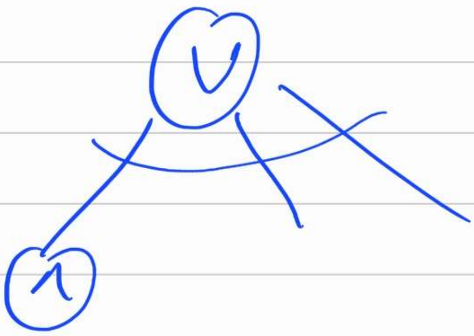
$$\bigvee_{x \in f^{-1}(1)} \left[\text{expression evaluating to } 1 \text{ on } x\right]$$

$f(1011) = 1$

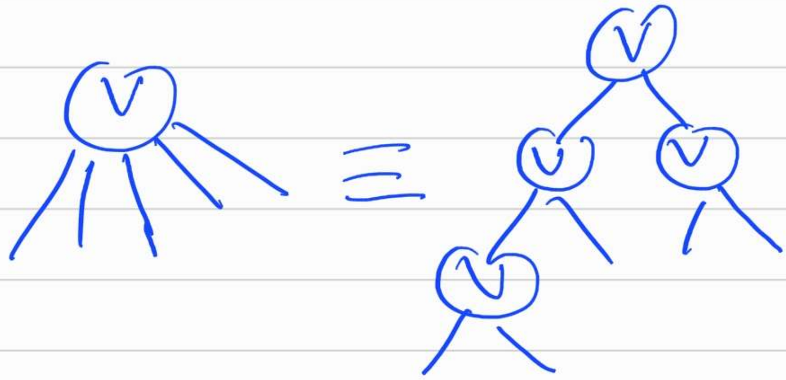
$x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4$

$\text{SIZE} \le |f^{-1}(1)| \cdot n \le 2^n \cdot n$

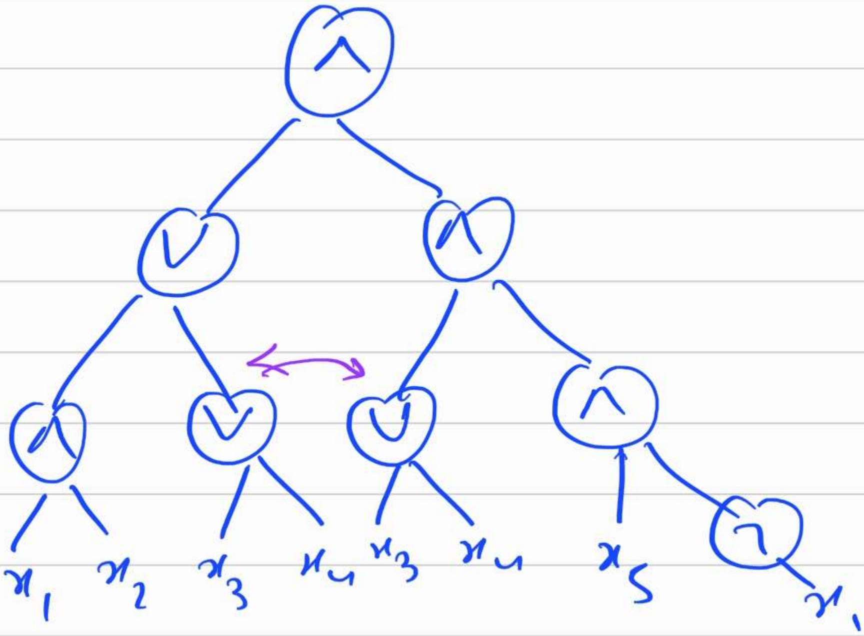
Formula vs Ckt :-



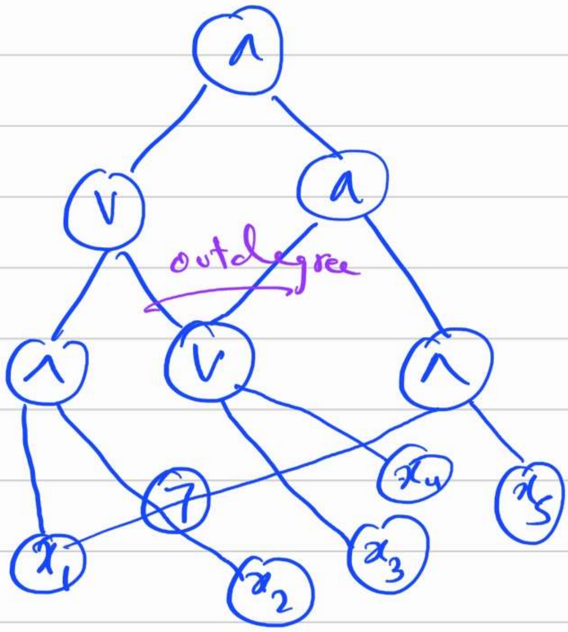
always reduce larger fan-in gates to indegree 2-gates



In formula every nodes has outdegree at most 1.
outdegree = fan-out



Tree



DAG

<u>Defn</u> :- P/poly : the class of languages that are decidable by polynomial

sized ckt family.

In other words : $P/_{Poly} := \bigcup_{c \geq 0} SIZE(n^c)$

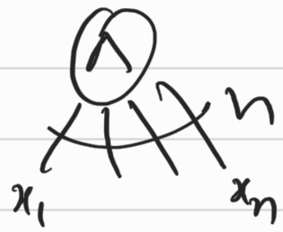
<u>Unary language</u> :- $L \subseteq \{1\}^*$

$$L = \{1, 111, 1111, 111111, \ldots\}$$

<u>Prop</u>:- Every Unary languange is in $P/_{Poly}$.

<u>Proof</u>:- $L^{=n} := L \cap \{1^n\} = \begin{cases} \{1^n\} & \text{if } 1^n \in L \\ \emptyset & o/w \end{cases}$

$\{C_n\}$ if $1^n \in L$,



o/w

0

<u>Note</u>! At every length ckts allow you to have a different algorithm.

This is in contrast to TM because there you have one algorithm for every input length.

UHALT := $\{ 1^n \mid$ binary encoding of n encodes $\langle M, x \rangle$ s.t. M halts on input $x \}$

Prop :- UHALT is undecidable.

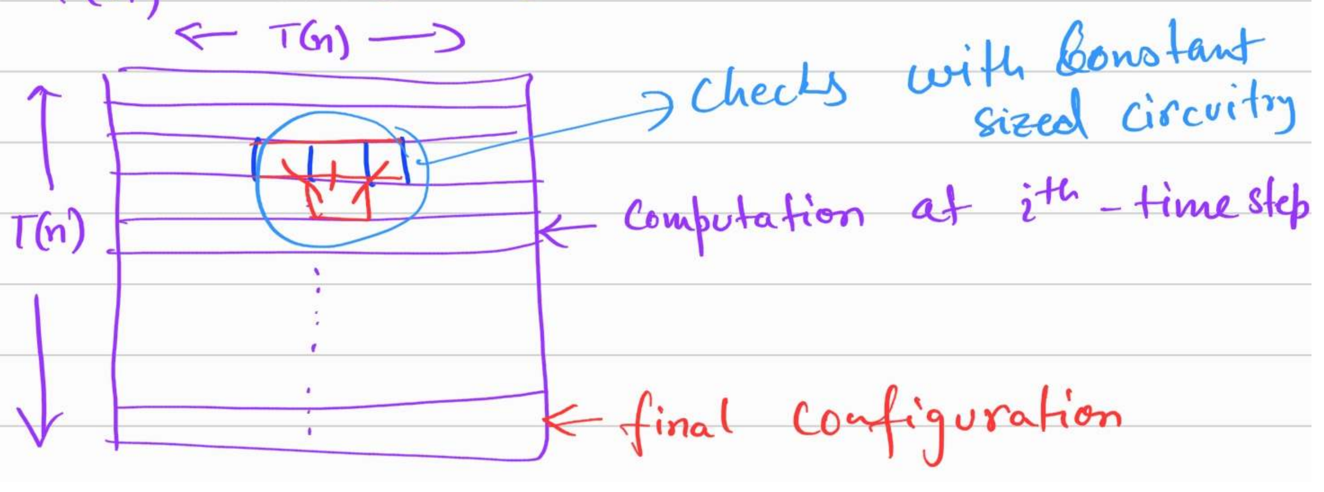
Proof :- Because HALTING Problem is undecidable.

BUT, UHALT $\in$ P/poly.

Thm :- $P \subseteq P/\text{Poly}$.

Proof :- (Cook-Levin Thm)

Let M be $T(n)$ - time DTM.



$\therefore$ you get a ckt of $O(T(n)^2)$.

<u>Also:</u> size of the ckt can be improved to $O(T(n) \log T(n))$ using Oblivious Turing Machines.

<u>NOTE:</u> you can produce this ckt in polynomial-time.

<u>P-Uniform Ckts:</u>

A ckt family $\{C_n\}$ is P-Uniform if $\exists$ a poly-time DTM M that on input 1^n produces the description of the ckt C_n.

$$M(1^n) \longrightarrow C_n$$

Turing Machines give Uniform family of ckts.

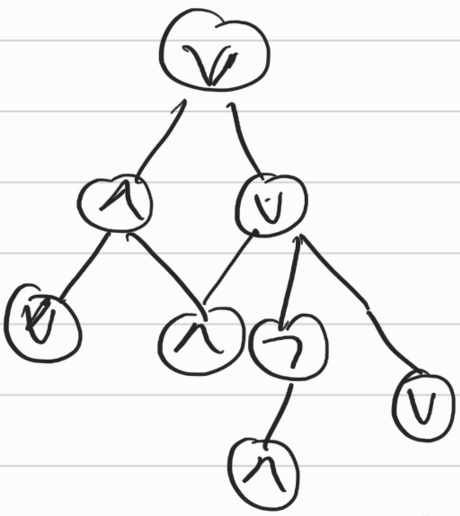
whereas Circuits by definition are allowed to be "non-uniform".

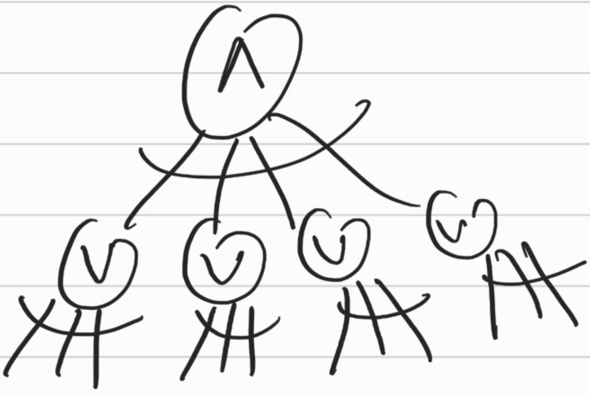
Defn:- (Ckt-SAT): Given a ckt C on n variables decide if $\exists$ an assignment $a \in \{0,1\}^n$ s.t. $C(a) = 1$.

You have seen: CNF-SAT

Conjunctive Normal Form

$(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\vee \cdot \cdot \vee) \wedge (\vee \cdot \cdot \vee)$





Lemma: Ckt-SAT $\in$ NP ✓

Lemma: Ckt-SAT is NP-Hard?

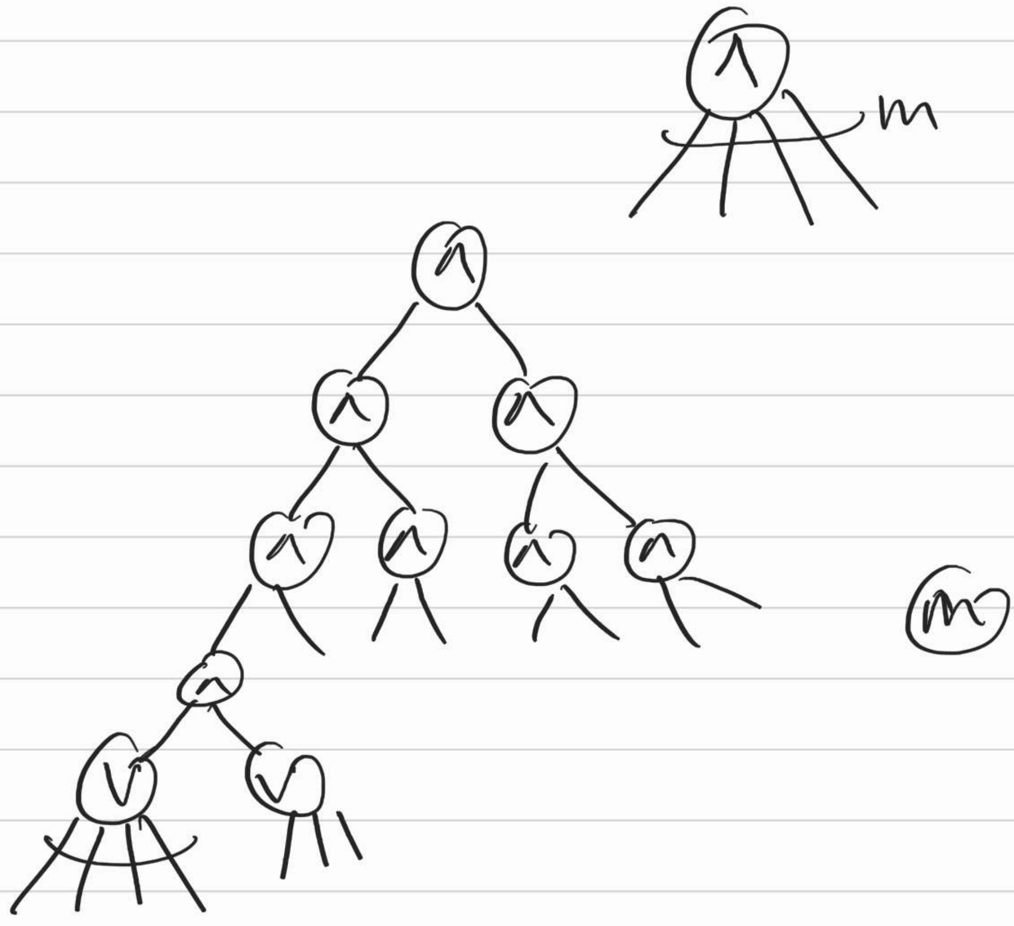
Ckt-SAT $\leq_p$ 3SAT (Exercise).

Does this prove Ckt-SAT NP-Hard?

No.

$3SAT \leq_P$ Ckt-SAT

$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_5 \vee \bar{x}_1 \vee x_{10}) \wedge (\ ) \wedge (\ )$



Thm :- Ckt-SAT is NP-complete.

P/poly :- class of poly-sized ckts.

Thm :- P = P-uniform P/poly

earlier we saw P ⊆ P/poly.

Requiring uniformity allows us to capture Turing Machines with Ckts.

What about the reverse direction?

Advice to TM:

TM that have a special read-only tape that has a string or "advice" on it.

Formally, for every input length n TM gets a string $\alpha(n)$ on the advice tape.

Usual Turing machine with a tape written $\alpha(n)$ on it.

Defn:- $DTIME(T(n)) / \alpha(n)$ is the class of languages decidable by a $T(n)$- time DTM M with $\alpha(n)$-bits of advice.

It means $\exists$ a sequence $\{a_n\}$ of strings with $a_n \in \{0,1\}^{\alpha(n)}$ and a TM M s.t.

$$x \in L \iff M(x, a_n) = 1.$$

Note:- for every input length n, $\exists$ a single string a_n.

$$P/\text{Poly} \equiv \bigcup_{\text{Thm: } c,d} \text{DTIME}(n^c)/n^d$$

$\Vert$

Ckts of poly size

Thm :- $P/\text{Poly} = \bigcup_{c,d} \text{DTIME}(n^c)/n^d$.

we saw that every Unary language is in P/poly.

Suppose we want a TM with an advice deciding the Unary language.

$\boxed{\forall\ x \in \{0,1\}^n \text{ has the same advice } a_n.}$

to decide a unary language.

for every length n

advice is 1 if $1^n \in L$

advice is 0 if $1^n \notin L$.

On input x

the machine M first figures out the length of x., say it is n.

if $a_n = 1$.

then M checks if $x = 1^n$

if $a_n = 0$.

then M rejects.

Thm :- $P/\text{Poly} = \bigcup_{c,d} \text{DTIME}(n^c)/n^d$

$\text{DTIME}(n^c)/n^d \subseteq P/\text{poly}$ (easy).

(poly-sized ckts)

$\Rightarrow$ Similar to Cook-Levin Thm.

fix the advice string in the ckt.

$L \in \text{DTIME}(n^c)/n^d$

$\Rightarrow \exists M$ and $\{a_n\}$ s.t. $a_n \in \{0,1\}^{n^d}$

s.t. $x \in L \iff M(x, a_n) = 1$.

Convert M into a ckt following Cook-Levin.

it takes two inputs x, a_n.

Then fix a_n corresponding to advice variables.

$$P/poly \subseteq \bigcup_{c,d} DTIME(n^c)/n^d$$

$L \in P/poly \Rightarrow \exists$ a $\{C_n\}$ s.t.

$x \in L \iff C_n(x) = 1$ when

C_n has polynomial size. $|C_n| \le s(n)$

So given ckt C_n and x you can evaluate C_n on x using a TM in polynomial-time.

So our advice to the TM is just the description of C_n for every n.

How many bits do you need to describe C_n?

At most $O(s(n)^2)$.

$UEVEN := \{1^n \mid n \text{ is even}\} \in P$

$UHALT := \{1^n \mid n \text{ ecodes } \langle M,x \rangle \text{ s.t. } M \text{ halts on } x\} \notin P$

$\in P/Poly$.

for an arbitrary unary language L

$|L \cap \{0,1\}^n| = 1$ or 0

$\therefore$ $L \in P/poly$.

Since $P/poly$ can decide undecidable languages

is it even worth studying ?

or, is all of NP in $P/poly$?

Thm [Karp-Lipton-Sipser Thm]

if $NP \subseteq P/poly$ then $PH = \Sigma_2^P$.

if $NP \not\subseteq P/poly \Rightarrow P \neq NP$