P vs NP: time complexity, verification, and nondeterminism

Wednesday, April 30, 2025 11:14 AM

- Verifying answers is easier than computing answers?

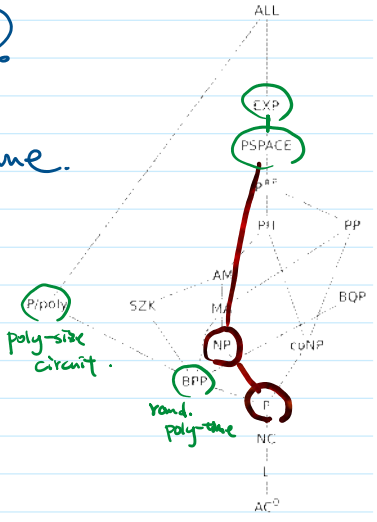
P : class of languages solvable by TM in poly-time.

NP : " verifiable "

formally. $P = \bigcup_{k \geq 0} TIME[n^k]$

$NP = \bigcup_{k \geq 0} NTIME[n^k]$.

nondeterministic



Open Question. $P = NP$?

- It's ridiculous that it is open.

- It's ridiculous that it is open.
- encapsulate the idea of "creativity"

- Why nondeterminism = verification?

Def. Verifier V of language L is a Turing machine/algorithm, s.t. $L = \{ \text{input} \in \Sigma^* : V \text{ accepts } \langle \text{input}, \text{proof} \rangle \text{ for some proof} \in \Sigma^* \}$

Prop. L accepted by NTM in poly-time iff L has verifier.

pf. "$\Leftarrow$": Use nondeterminism to guess the proof.

$N(w)$: «input w of length n»

1. nondeterministically guess proof of length $n^{O(1)}$
2. run V on $\langle w, \text{proof} \rangle$. return accordingly.

"$\Rightarrow$": Simulating NTM, the accepting branch being the proof.

$V(w, \text{proof})$:

1. simulate $N(w)$, treat each symbol in proof as nondeterministic choice at each branch.
2. accept/reject based on the branch.