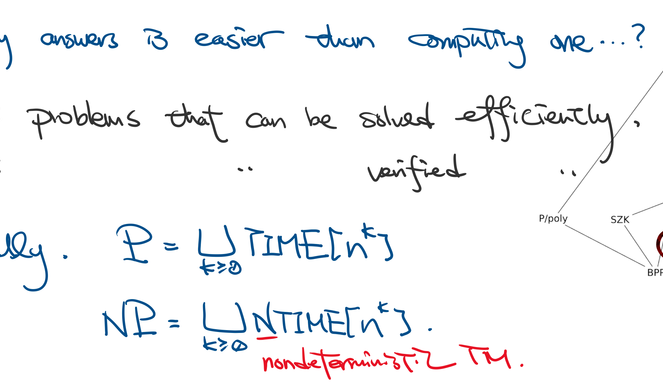
P vs NP: time complexity, verification, and nondeterminism

2/8/21 10:12 AM

~~Administrivia~~.

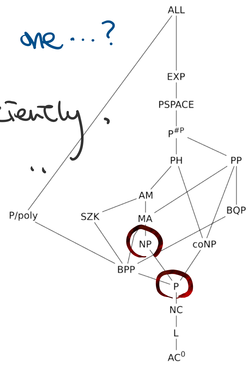
- HW4 out (due 2/12)
- Midterm 1 tomorrow (2/9)
  - closed-book exam. everything about regular languages.
  - will be posted online at 12am / today midnight
  - submit before tomorrow midnight / 11:59pm
  - schedule 15-min oral exam w/ me.
- CS31 spring'21 looking for TAs!



- Verifying answers is easier than computing one...?

P : problems that can be solved efficiently.

NP : " verified "



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

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

nondeterministic TM.

Open Question. $P = NP$?

- 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. $L = \{w \in \Sigma^* : V \text{ accepts } \langle w, proof \rangle \text{ for some } proof \in \Sigma^*\}$

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

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

"$\Rightarrow$": Simulating NTM, make decision based on symbols in proof. ∎

Q. What's the next question?

- space complexity.
- EXP.
- why polynomials as def?
- is $NP \cap coNP = P$?