L23 2022-11-10

Simulating nondeterminism.

def $P = \bigcup_c DTIME(n^c)$ $NP = \bigcup_c NTIME(n^c)$

$EXP = \bigcup_c DTIME(2^{n^c})$ $NEXP = \bigcup_c NTIME(2^{n^c})$.

th $NP \subseteq EXP$

proof: try all possible witnesses.

Formally: Let $L \in NP$, V verifier for L running in time n^c.

Let d st $x \in L$ iff $\exists y, |y| \leq |x|^d$, $V(x,y)$ accepts.

M: for each string y of size $\leq |x|^d$:
 ∟ if $V(x,y)$ accepts: accept
 reject.

M accepts x iff $\exists y$ --- $\Rightarrow$ M is correct.

M runs in time $O(2^{n^d} \cdot n^c)$, i.e. exponential.

& we can prove $NTIME(f) \subseteq DTIME(2^{O(f)})$ similarly, as long as f is efficiently computable.

Q: is there a way to simulate nondeterminism other than brute-force?

We think of TMs running in poly-time as efficient, and exp time as inefficient.

We have $P \subseteq NP \subseteq EXP \subseteq NEXP$.

Also, $P \neq EXP$. But is $P = NP$?

Poly-time reductions

def. $A \leq_p B$ if $\exists f: A \to B$ poly-time computable st $x \in A$ iff $f(x) \in B$.

prop. $A \leq_p B$ and $B \in P \Rightarrow A \in P$
$NP \Rightarrow NP$.

e.g. $SAT \leq_p 3SAT$

Given CNF over $x_1 - x_n$ and clauses $C_1 - C_m$, we build CNF where

$$C_i = (\ell_1 - \ell_k) \mapsto (\ell_1 \vee y_{i1}) \wedge (\bar{y}_{i1} \vee \ell_2 \vee y_{i2}) \wedge \dots \wedge (y_{ik-1} \vee \ell_k).$$

Easy to check poly-size and sat iff orig is sat.