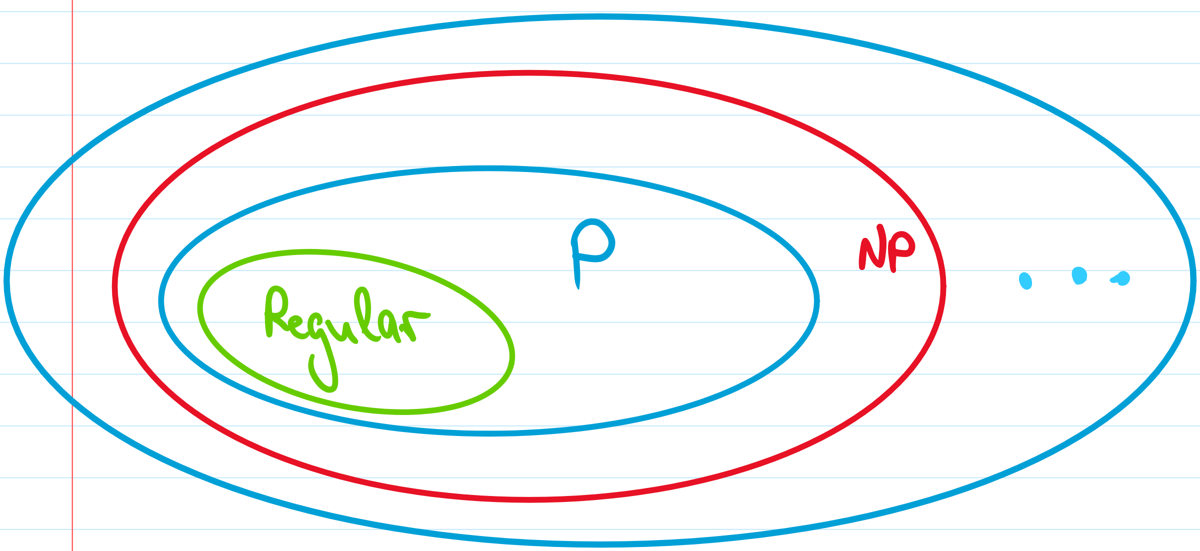
Lecture 19

Friday, June 28, 2019 11:27 AM

Complexity Theory



- A_{TM}
- Truth

. . .

Decidable

DTM M

on x

$$\left.\begin{array}{l} \bullet\ \text{config}_0 \\ \downarrow \\ \bullet\ \text{config}_1 \\ \downarrow \\ \vdots \\ \downarrow \\ \bullet\ \text{config}_t \end{array}\right\} \text{time } t$$

$$\text{Time}_M(n) = \max_{x:\,|x|=n} \{ \text{time}_M(x) \}$$

$$\mathrm{Time}_M(n) = \max_{x : |x| = n} \{ \mathrm{time}_M(x) \}$$

worst-case time measure
over all inputs of size n

For $t : \mathbb{N} \to \mathbb{N}$,
define $\mathrm{TIME}(t) = \{ L \mid \exists$ DTM M deciding L s.t. $\mathrm{Time}_M(n) \le O(t(n)) \}$

Define
$$P = \bigcup_{k>0} \mathrm{TIME}(n^k)$$

"Efficient algo = Polytime algo"

Why we like P:

(1) P is robust: other models of computation (k-tape TM, RAM, etc.) can be simulated by a 1-tape TM with poly slowdown only

slowdown only.

(2) P is closed under polytime reductions:

if $B \in P$ & $A \leq_p B$

(A is polytime reducible to B)

then $A \in P$.

(3) P contains many natural, efficiently solvable problems.

Why we ignore constant factors when defining Time Complexity:

TM M, runtime $t(n)$

$\Downarrow$ $\forall$ const $c > 1$

TM M', run time $n + \frac{t(n)}{c}$

s $L(M') = L(M)$

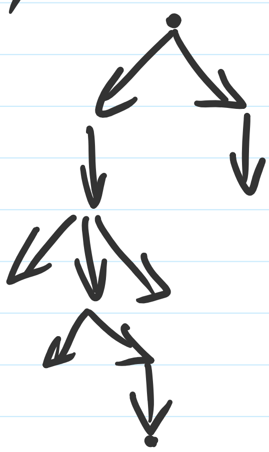
c

& $L(M') = L(M)$.

(<u>Idea</u>: "compress" each c-tuple of symbols of Γ of M
into a single super-symbol of Γ' of M',
i.e., define $\Gamma' = \Gamma^c$.
Then M' can simulate $\approx c$ steps of M
in 2 steps.)

Nondeterministic PolyTime (NP)

NTM M
on input x



depth = time of M on x

$$\mathrm{NTime}_M(n) = \max_{x : |x| = n} \{ \mathrm{time}_M(x) \}$$

For $t : \mathbb{N} \to \mathbb{N}$,

$$NTIME(t) = \{ L \mid \exists \text{ NTM } M \text{ for } L \text{ with } NTime_M(n) \le O(t(n)) \}$$

$$NP = \bigcup_{k>0} NTIME(n^k)$$ ← 1st defn of NP

2nd defn of NP (equivalent)

L:

x $\quad |x| = n$

Prover $\xrightarrow{\text{"proof" } \pi}$ Verifier

"proof" that x is a Yes-instance

$V \in P$

$$V(x, \pi) \stackrel{\text{Want}}{=} \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{o.w.} \end{cases}$$

$L \in NP$ if $\exists V, \exists c > 0$ s.t.

$\forall x,$

(1) $x \in L \Rightarrow \exists \pi \quad |\pi| \le |x|^c$

(1) $x \in L \Rightarrow \exists\ \pi,\ |\pi| \le |x|^c,$

$$V(x, \pi) = 1$$

(2) $x \notin L \Rightarrow \forall\ \pi,\ |\pi| \le |x|^c,$

$$V(x, \pi) = 0$$

Ex: Composite $= \{ \langle n \rangle \mid n \text{ is composite} \}$

Composite $\in$ NP

x

Prover $\xrightarrow[(d \ne 1,\ d \ne x)]{\text{divisor } d \text{ of } x}$ Verifier will check d divides x

Defn 1 of NP $\equiv$ Defn 2 of NP

$\Rightarrow$

NTM polytime TM M

for language /

for language L

x, $\Longrightarrow$ π = "accepting branch of M on x"

$\Longleftarrow$

NTM M on x:

"Nondet. guess a π,

Run $V(x, \pi)$."

Examples

SAT:

Given: formula $\varphi(x_1, \dots, x_n)$

Decide: is φ satisfiable?

$$(x \vee \bar{y}) \wedge (\bar{x} \vee z \vee \bar{w}) \wedge \bar{y}$$

SAT $\in$ NP:

π = "assignment to φ"

π = "assignment to φ"
(satisfying)

Search-to-Decision :

SAT-Search : Given $\varphi(x_1, \ldots, x_n)$,
Find a satisfying ass't
if exists.

SAT-Search $\leq_p$ SAT