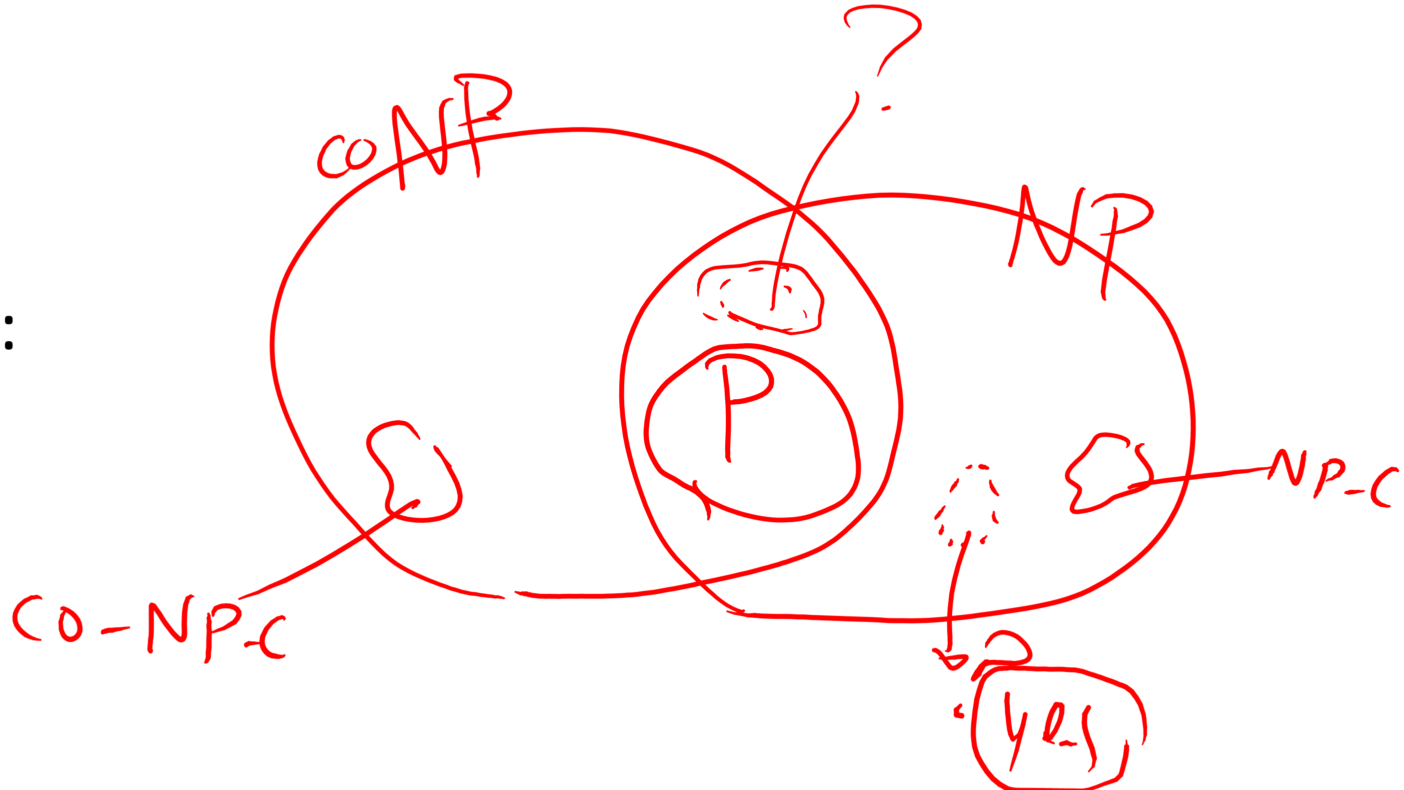
Computational Complexity

Mohammad Mahmoody

Session 8
6 Feb 2014

Last Time

- **coNP** consists of all L such that $\overline{L} \in \mathbf{NP}$
- $\mathbf{coP} = \mathbf{P}$
- Common belief in picture:



Defining NP, coNP using quantifiers

$L \in NP$: $\boxed{\begin{array}{l}\exists \text{ Verifier } V_L \ (\text{polynomial time}) \\ \exists\, p(\cdot) : \text{polynomial.}\end{array}}$ s.t. $x \in L$ iff $\exists\, w \in \{0,1\}^{p(n)}$ s.t. V_L accepts (x, w)

$n = |x|$

$\bar{L} \in coNP$: $\boxed{\begin{array}{l}\exists \text{ Verifier } V_L \\ \exists\, p(\cdot) \ldots\end{array}}$ $x \in \bar{L}$ iff $\forall\, w \in \{0,1\}^{p(n)}$ V_L ~~reject~~ accept (x, w)

Defining **NP** using
Non-**D**eterministic **T**uring **M**achine (NDTM)

NDTM: $\exists$ two transition functions $\boxed{\delta_1, \delta_2}$

$L \in NP$: $\exists$ NDTM (poly-time) M_L s.t. $x \in L$ iff

M_L "accepts" x

($\exists$ a way that M_L accepts)

$L \in$ Co-NP: $\exists$ NDTM M_L s.t. $x \in L$ iff.

~~for~~ $\forall$ branches of $M_L(x)$ we get reject.

Equivalence of Definitions

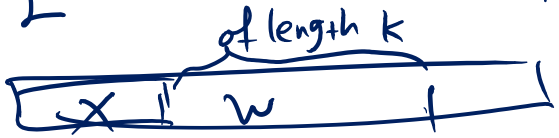
of class NP.

Def$_1$ (witness-based def) $\equiv$ D$_2$ (NDTM)

$\underline{L \in NP}_1$ (def based on witness verif).

$\overset{?}{\Rightarrow}$ $L \in NP_2$ (def ~~~ NDTM).

$\exists$ V_L s.t. $x \in L$ iff $\exists w$ s.t. (x, w) accepted by V_L



read input.

go into state q

for k steps. generate w_i non-deterministically

after k steps w is generated

Run V_L over (x, w).

Shows. $NP_1 \subseteq NP_2$

claim $NP_2 \subseteq NP_1$

witness = choices made

Today

- Time Hierarchy Theorem: “more time $\rightarrow$ more power”
- Diagonalization
- Limits of Diagonalization

Time Hierarchy Theorem

- Informal: in any "reasonable" model of computation (Turing Machine, RAM, etc.) having more "time" implies more "computational power".

$P = \bigcup_{k \in \mathbb{N}} DTime(n^k)$

- Formal: Let $\boxed{\mathbf{Dtime}(T(\cdot))}$ be the set of languages that are decidable in time $T(n)$ by a Turing machine. Then for all natural k:

$$\mathbf{Dtime}(n^{k+1}) \overset{\supseteq}{\neq} \mathbf{Dtime}(n^k)$$

$L \in Ntime(T(\cdot))$ iff $\exists$ NDTM of time $T(\cdot)$ that accepts L.

True $\longrightarrow$ $Ntime(n^{k+1}) \neq Ntime(n^k)$

Recalling the proof of Halting Theorem

- Definition of Halting Language:
  HL = $\{M \mid M$ (as Turing Machine) halts over $M(as\ input)\}$
- Halting Theorem: **HL** is not decidable

Assume A Solves HL

Look at this machine M : takes x input.

if $x \in HL$: fall in loop

$x \notin HL$: halt

What happens if run M over M

either
$\begin{cases} M \text{ halts over } M \longrightarrow M \in HL \Longrightarrow M \text{ does not halt on } M \\ M \text{ dos Not halt over } M \Rightarrow M \notin HL \Longrightarrow M \text{ halts on } M \end{cases}$

$\text{Ptime}(n^2) \not\supseteq \text{Dtime}(10n)$

Proving $\mathbf{Dtime}(n^{k+1}) \neq \mathbf{Dtime}(n^k)$

- Recall Universal TM:
  UTM gets (M, x) as input, simulates T steps of $M(x)$ in time $T \log T$

Construct language $L \in \text{Dtime}(n^2)$ but $L \notin \text{Dtime}(10n)$

def of L: given x: if run x over x for time $n^{1.5}$ t

$\begin{cases} \text{if } x \text{ accepts } x \Longrightarrow \text{reject}: x \notin L \\ \text{oth.} \Longrightarrow x \in L \end{cases}$

claim 1

$\rightarrow$ $L \in \text{Dtime}(t \log t) \approx \text{Dtime}(n^{1.5} \underbrace{\log(n^{1.5})}_{= \log(n)}) \subseteq \text{Dtime}(n^2)$

$\boxed{\text{claim 2}}$ $L \notin \underline{\text{Dtime}(10n)}$; $\exists M$ solves L in $10n$

Run M over M $\begin{cases} \text{accep} \rightarrow \text{reject} \\ \text{rej.} \end{cases}$

Why is it called "Diagonalization" ?