CSS.203.1
Computational Complexity
– Lecture #4 (24 Feb,'21)
Instructor: Prahladh Harsha

Today

- Cook-Levin Theorem
- Decision vs Search
- coNP, NEXP
- Thoughts on NP, $NP \overset{?}{=} P$ ....

Cook-Levin Theorem: SAT is NP-complete

$\forall L \in NP, \quad L \leq_p SAT$ (need to show).

Let $L \in NP$.

ie $\exists$ a TM M & two poly p, q s.t

$x \in L \iff \exists u \in \{0,1\}^{p(|x|)}, \; M(x,u) = 1$

& furthermore M runs in time $q(|x|)$ on i/p (x,u).

Assumptions: (on M):

(i) M - 2-tape TM

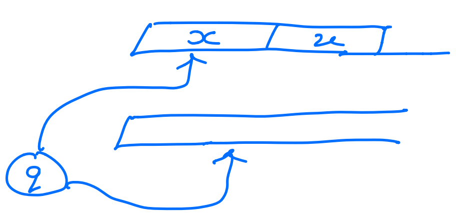
(ii) M is oblivious

Notation:

M's input $y = \langle x, u \rangle$

$T = q(|x|)$

– Snapshot



$z = (q, a, b)$

Snapshot: View of the head of TM

$z \in Q \times \Gamma \times \Gamma$

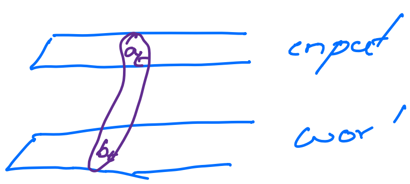
For each time step t

$prev(t)$:= Previous time step t' at which the head of TM is in the same position on the work tape as at time t (& if this is first 0)

$input\text{-}pos(t)$:= Location of head on input tape at time t.

<u>Obliviousness of TM</u>: $prev(t)$ & $input\text{-}pos(t)$ are only fns of $|x| = n$ & not x.

(Can compute $prev(t)$ & $input\text{-}pos(t)$ for all $t \leq Q(|x|)$ by running M on a trivial i/p $\underbrace{0^{|x|}}_{x} : \underbrace{0^{p(|x|)}}_{u}$)



Z_t - snapshot at time t

$Z_t = (q_t, a_t, b_t)$

q_t

$$q_t := \delta(Z_{t-1})\big|_1 \qquad \delta : Q \times \Gamma^2 \to Q \times \Gamma \times \{L, R\}^2$$

$$a_t := y_{input\text{-}pos(t)}$$

$$b_t := \delta(Z_{prev(t)})\big|_2$$

$$Z_t = F(Z_{t-1}, y_{input\text{-}pos(t)}, Z_{prev(t)})$$

$Z_0 \mapsto Z_1 \mapsto Z_2 \mapsto \cdots \mapsto Z_T$

What do we need to check that the above is a valid sequence of snapshots when the m/c M is run on i/p (x, u) for some $u \in \{0,1\}^{q(|x|)}$.

1. Initial Checks

(a). First n-bits of y are equal to x $\} O(n)$

(b) Initial snapshot Z_0 is initialized correctly $\} O(1)$

2. Transition Checks

$\forall t \in \{2, \ldots T\}$

$Z_t = F(Z_{t-1}, y_{\text{inputpos}(t)}, Z_{\text{prev}(t)})$ $\} \forall$ time t φ_t

$f : \{0,1\}^{3c+1} \to \{0,1\}$

c = #bits reqd to encode a snapshot

3. Final Acceptance Check

Final snapshot is accepting. $\}$ — $O(1)$

Encode the above as a CNF φ_x

Variables $(\varphi_x) := Z_0 Z_1 \ldots\ldots Z_T$

$= y_1 \ldots \quad y_{p(|x|)}$.

Total size of the CNF constructed

$= O(n) + O(1) + T \cdot O(1) + O(1)$

$= O(T + n)$

$x \in L \iff \varphi_x$ is satisfiable.

Key (to proof):

Each step of computation is local (constant-size).

n - clause

$$\varphi_x = \bigwedge (\text{initial step}) \wedge \bigwedge_{i=1}^{T} (\text{Transition check}) \wedge (\text{Final step})$$

Remarks:

1. $L \leq_p SAT$ for every $L \in NP$
   - Reduction from $L \in NTIME(T(n))$ to SAT
     - Reduction runs in time $O(T \log T)$
     - Size of $|\varphi_x| = O(T \log T)$

2. $L \leq_p 3SAT$

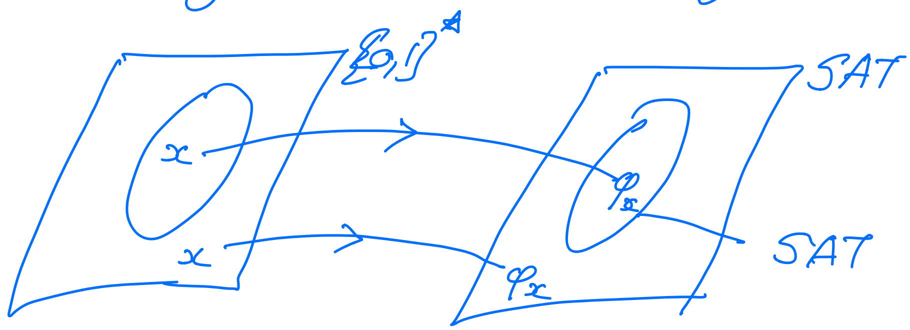
   $x \longmapsto \varphi_x$

   $u \longleftrightarrow a$

   satisfying witness — satisfying assignment

   1-1 mapping between satisfying witnesses of x & satisfying assignments of φ_x

Parsimonious Redn: poly time redn between problems in NP (such that there is a bijection between satisfying witnesses of the source & target instances)



3SAT is NP-complete

Redn: SAT $\leq_p$ 3SAT.

$$\underline{\Phi} \longmapsto \underline{\psi}$$

CNF $\qquad$ 3CNF (ie every clause has at most 3 vars)

$$\underline{\Phi} = C_1 \wedge C_2 \ldots \wedge C_m$$

$$\downarrow$$

$$\psi_1 \wedge \psi_2 \quad \ldots \quad \wedge \psi_m$$

ψ_i - 3CNF formula.

$$C_i \longmapsto \psi_i.$$

$\#\text{lits}(C_i) \leq 3 \implies \psi_i = C_i.$

$\# \text{lits}(C_i) \geq 4$

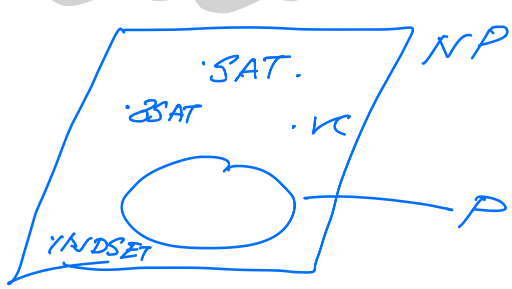
$$C_i = (\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4)$$

$$\psi_i = \begin{cases} (\ell_1 \vee \ell_2 \vee z) \\ (\bar{z} \vee \ell_3 \vee \ell_4) \end{cases}$$

$$C_i = (\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5)$$

$$\psi_i = \begin{cases} (\ell_1 \vee \ell_2 \vee z_1) \\ (\bar{z}_1 \vee \ell_3 \vee z_2) \\ (\bar{z}_2 \vee \ell_4 \vee \ell_5) \end{cases}$$

Search vs Decision

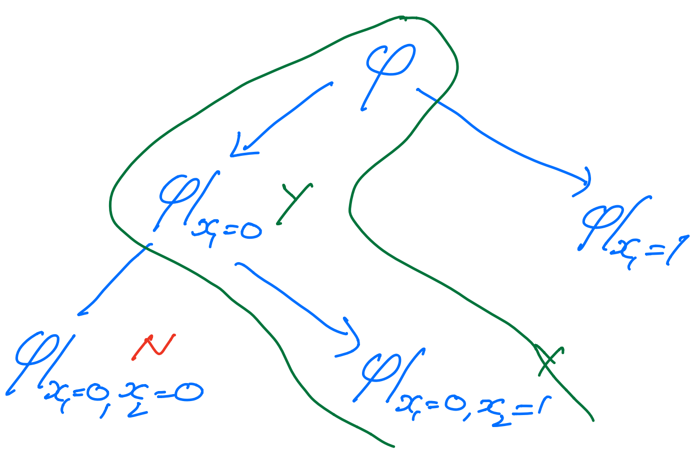


Qn: P = NP

$\Downarrow$

then can check if a formula is satisfiable.

Can we also find a satisfiable assignment if one exists in ptime?



Downward Self Reducibility of SAT.

$$T(n) \leq 2T(n-1) + O(1)$$

Thm: If P=NP, then there exist a polynomial time algorithm for every lang in NP that when given an instance x, checks if $x \in L$ & if so finds a satisfying witness.

—

coNP: co-nondeterministic.

$$coNP = \{ L \mid \bar{L} \in NP \}$$

(not complement of NP)

coNP: $L \in coNP$, if $\exists$ TM M & poly p, q s.t

$x \in L \iff \forall u \in \{0,1\}^{p(|x|)}$, $M(x,u) = 1$

& M runs in time $q(|x|)$ on i/p (x,u).

NP - short proofs of membership

coNP - short proofs of non membership

eg: $TAUT = \{ \varphi \mid \varphi$ is always true$\}$

eg: $x \vee \bar{x}$

$\overline{TAUT} = \{ \varphi \mid \varphi$ is false for some assignment$\}$

NEXP:

$NP = \bigcup_{c:c\geq 1} NTIME(n^c)$; $P = \bigcup_{c:c\geq 1} DTIME(n^c)$

$NEXP = \bigcup_{c:c\geq 1} NTIME(2^{n^c})$ $EXP = \bigcup_{c:c\geq 1} DTIME(2^{n^c})$

eg:- $TMSAT = \{ (\alpha, x, 1^n, 1^t) \mid \exists u \in \{0,1\}^n$ st $M_\alpha(x,u) = 1$ & M_α runs in time $t \}$

TMSAT is NP-complete.

$TMSAT_{\text{-binary}} = \{ (\alpha, x, \underbrace{\lfloor n \rfloor, \lfloor t \rfloor}_{\text{binary}}) \mid \ldots \}$

$\in NEXP$
$\in NEXP\text{-hard}$ } NEXP-complete.

Thm: $P = NP \Rightarrow EXP = NEXP$

Pf: Padding Technique

Assume $P = NP$.

$L \in NEXP$

$L \in NTIME(2^{n^c})$ for some constant c.

$L_{pad} = \{ (x, 1^{2^{|x|^c}}) \mid x \in L \}$

$L_{pad} \in NTIME(n) \in NP \underbrace{\subseteq}_{\text{assumption}} P$

What can you say about L?

- On input x
  - pad it
  - run the ptime alg for L_{pad}

$L \in EXP$

Hence, $NEXP = EXP$. $\boxtimes$

<u>Padding</u>: $\begin{cases} \text{Collapses} & \text{scale up} \\ \text{Separations} & \text{scale down} \end{cases}$

Thoughts of NP, $NP \overset{?}{=} P$, $NP \overset{?}{=} coNP$

① $P \neq NP$ question.

Computational version of

"Can ingenuity be automated?"

② . $NP \overset{?}{=} coNP$ question.

$\boxed{TAUT}$ If $NP = coNP$, then every $\varphi \in TAUT$ has a short proof that it is a tautology.

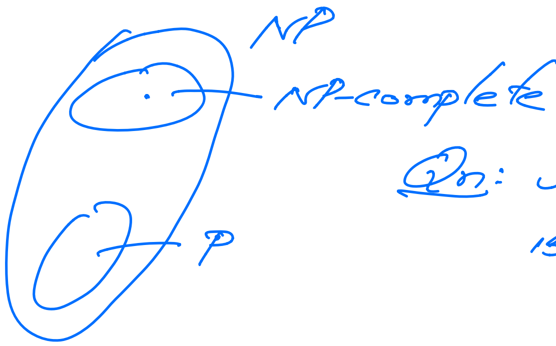
Short proofs of membership

$\overset{?}{\parallel}$

Short proofs of non-membership

Proof Complexity

(3)



Qn: If $NP \neq P$, then is any $L \in NP \setminus P$ NP-complete?

No: NP-intermediate problems.

(Ladner's Theorem).

(4) What if $P = NP$?

Algorithmicist's Utopia.

- Fine grained complexity.
- Crytographer's nightmare.

(5) Coping w/ NP-hardness

- heuristics
- approximation algorithms

Next time: Diagonalization