Today

- Polynomial time Reductions
- NP-Completeness
- Cook-Levin Theorem
- Dec vs Search
- coNP, NEXP.

CSS.203.1 Computational Complexity
– Lecture #3 (22 Feb,'21)
Instructor: Prahladh Harsha

Last time:

NP Examples: SAT, INDSET
Composite, TSP, ..

Reductions

Polynomial time Karp Reductions.

L_1, L_2 - be two languages

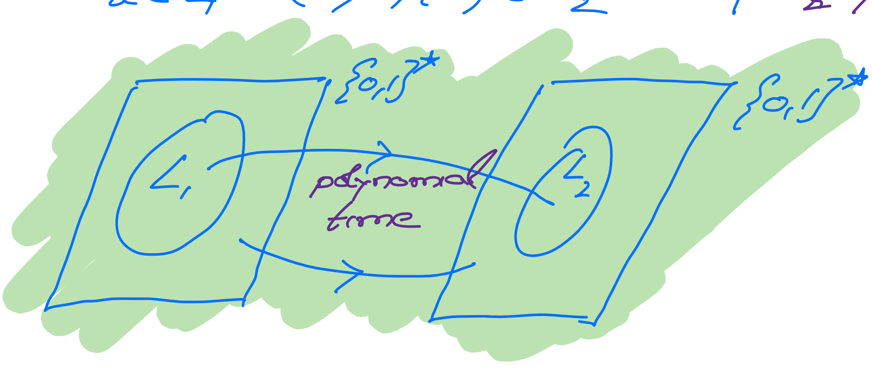
$L_1 \leq_p L_2$ (L_1 is polytime reducible to L_2)

if

$\exists$ polytime computable fn
$f : \{0,1\}^* \to \{0,1\}^*$

$x \in L_1 \iff f(x) \in L_2$

i.e., $\exists$ a det. TM M
& polynomial p
s.t.
$f(x) = M(x)$
& M runs in time
at most $p(|x|)$
on i/p. x.



Suppose $L_2 \in P$ given by TM N

What is the alg for L_1.

N' = On input x

1. Compute $f(x)$
2. Run N on $f(x)$ & output the answer.

NP-hard:

L is NP-hard if $\forall L' \in NP$, $L' \leq_p L$

$$\left.\begin{array}{l} L \in NP \\ L - \text{NP-hard} \end{array}\right\} \Rightarrow L - \text{NP-complete}$$

Example of NP-complete problem:

$TMSAT = \{ (\alpha, x, 1^n, 1^t) \mid \alpha$ is a NTM & $\exists u \in \{0,1\}^n$, st $M_\alpha(x,u) = 1$ in time t steps $\}$.

<u>Observation</u>: $L \in NP \Rightarrow L \leq_p TMSAT$

<u>Pf:</u> $L \in NP$

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

$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$ s.t $M(x,u) = 1$ & M runs in time $q(|x|)$.

$f: \quad x \longmapsto (\langle M \rangle, x, 1^{p(|x|)}, 1^{q(|x|)})$

Conclusion: TMSAT is NP-hard, TMSAT $\in$ NP $\Big\}$ $\Rightarrow$ TMSAT is NP-complete.

Prove a different problem is NP-complete.

Cook-Levin Theorem:

(1) SAT is NP-complete

(2) 3SAT is NP-complete

φ - Boolean formula - CNF format

$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$

where each C_i - clause

- disjunction of literals

$C_i = l_{i1} \vee l_{i2} \vee \ldots \; l_{im_i}$

l_k - x or $\bar{x}$.

SAT $= \{\varphi \mid \varphi$ is a CNF formula that is satisfiable$\}$

φ is 3CNF if each clause has ≤ 3 literals

3SAT $\subseteq$ SAT when you restrict to 3CNF formulae.

Properties of ptime Reductions.

Lemma:

(1) $L_1 \leq_p L_2$ & $L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$ (Transitive)

(2) L is NP-hard & $L \in P \Rightarrow NP \subseteq P$

(3) L is NP-complete

($L \in P \iff NP \subseteq P$)

3SAT is NP-complete.

Eg 1: 0/1 - Linear Programming.

Instance:

Linear Inequalities

$$\left.\begin{array}{l} x_1 + x_3 + x_5 \geq 10 \\ x_1 - x_2 \geq 8 \\ \vdots \end{array}\right\} m \text{ inequalities}$$

$x_i \in \{0,1\}$

Qn: Are they satisfiable?

IP $= \{ \psi \mid \psi$ is a satisfiable 0/1 - Integer programming instance $\}$

$3SAT \leq_p IP$.

$\varphi \xmapsto{f} \psi$

$(x \vee y \vee \bar{z}) \longrightarrow x + y + (1 - z) \geq 1$

$x, y, z \in \{0,1\}$

$$\left.\begin{array}{l} IP \text{ is NP-hard} \\ IP \in NP \end{array}\right\} \Rightarrow IP \text{ is NP-complete.}$$

$\boxtimes$

Eg 2: Independent Set Problem

$INDSET = \{(G, k) \mid \exists$ an ind set of size $\geq k$ in the undirected graph $G\}$

$INDSET \in NP$

Lemma: $3SAT \leq_p INDSET$ (Karp)

Pf:

$$\varphi \xmapsto{f} (G_\varphi, k_\varphi)$$

$\varphi \in 3SAT \iff \exists$ an indset of size k_φ in G_φ.

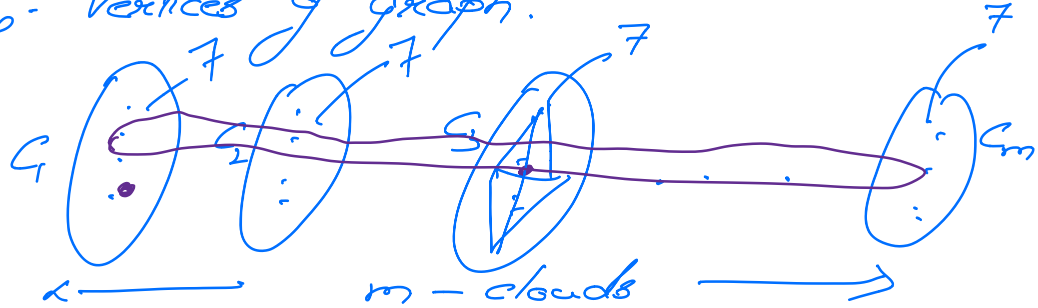
$\varphi = C_1 \wedge \ldots \wedge C_m$

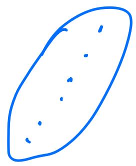
$C_i = (x_{i_1} \vee x_{i_2} \vee \overline{x_{i_3}})$

$G_\varphi = (V_\varphi, E_\varphi)$.

$|V_\varphi| = 7m$

V_φ: Vertices of graph.





C_i – 7 vertices

- refer to the ~~se~~ 7 satisfying assignments in Clause C_i

$V_\varphi = \bigcup_{i \in [m]} \{\text{satisfying assign to } Var(C_i)\}$.

(i, a) ↳ clause index; $a : Var(C_i) \to \{0,1\}$ & is satisfying for C_i.

$E_\varphi := (i,a) \sim (j,b)$

if (i,a) & (j,b) refer to inconsistent assignments.

Obs: Each cloud is a clique.

$k_\varphi = m$ (m – # clauses).

$\varphi \longmapsto (G_\varphi, k_\varphi)$

$\varphi \in SAT \iff \exists$ an indset of size $\geq m = k_\varphi$ in G_φ.

$\boxtimes$

Cor: INDSET is NP-complete.

Next: Proof of Cook-Levin Theorem.

CNF formulae:

Universality of CNF formulae:

Obs: $f : \{0,1\}^\ell \to \{0,1\}$, then there exists a CNF formula φ_f s.t.

$$\varphi_f(a) = f(a), \quad \forall a \in \{0,1\}^\ell$$

Furthermore, φ_f has at most 2^ℓ clauses with ℓ literals each.

$$(|\varphi_f| \leq \ell \cdot 2^\ell).$$

$$f(x_1 = 1, x_2 = 0) = 0$$

$$(\bar{x}_1 \vee x_2)$$

—

$x, y \in \{0,1\}^\ell$ ℓ-bit strings.

$(y_1 \Rightarrow x_1)$ $(x_1 \Rightarrow y_1)$

$x \equiv y$: $(x_1 \vee \bar{y}_1)$ $(\bar{x}_1 \vee y_1)$

$\vdots$

$(x_\ell \vee \bar{y}_\ell)$ $(\bar{x}_\ell \vee y_\ell)$

} 2ℓ-clause formulae that capture equality.

—

$L \in NP$ $L \leq_p SAT$

$L \in NP$ i.e. $\exists$ a TM M & poly p, q s.t. $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}, M(x, u) = 1$ &

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

$$x \longmapsto \varphi_x$$

$x \in L \iff \varphi_x \in SAT$

Given x, $\psi_x : \{0,1\}^{p(|x|)} \longrightarrow \{0,1\}$

$$u \longmapsto M(x,u)$$

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

↳ has a CNF representation

Say $\varphi_{(\psi_x)}$

Issue: Redn does not run in ptime since φ_{ψ_x} can be exponentially large.

Haven't used:

ψ_x : computable by a polynomial time TM.

$\overline{M}$ – assumptions

(i) 2 tapes

(ii) oblivious

$\Big\}$ $q(n) \longrightarrow O(q(n) \cdot \log q(n))$

CNF - formula

- Conjunction of clauses
- $\bigwedge_{i=1} \Big(\qquad \Big)$
- $\bigwedge_{i}$ (Step i is obtained correctly from Step $(i-1)$)
  $\wedge$ (Step 1 is initialized properly)
  $\wedge$ (Final Step is accepting)

<u>Snapshot</u>: View of the head of TM at a particular instant of time

$\cap$

$Q \times \Gamma \times \Gamma$

$z = (q, a, b)$

↳ head is in state q

↳ tape 1 has letter a under head

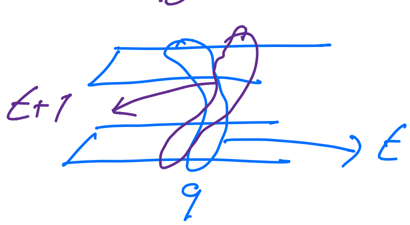
tape 2 " " b " "

$y = x \circ u$ (input for TM M)

$z_1 \to z_2 \to z_2 \ldots \to z_{q(|x|)}$

$(q_s, \triangleright, \triangleright)$

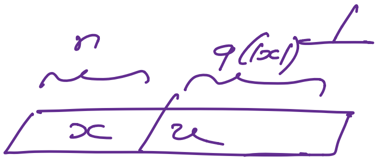
$z_t = F(z_{t-1}, z_{prev(t)}, y_{input\text{-}pos(i)})$



$z_{t+1} = (q_{t+1}, a_{t+1}, b_{t+1})$

$z_t = (q_t, a_t, b_t)$

$prev_i(t)$ — previous time at which TM M was on tape 2 at the same position as the ~~ear~~ t.
or
$\perp$



$input\text{-}pos(t)$: Position on input tape at time t.

Observation: $prev(t)$ & $input\text{-}pos(t)$ are fns only of $|x|$ & not x.
(because M is oblivious)

(Can compute these fns by just running M on i/p $0^{|x|}$.)