CPSC 421/501, Nov 17, 2020

- Define: NP-complete

- SUBSET-SUM
- VERTEX-EXPANSION

} are NP-complete

CPSC 501: Please fill out Canvas "discussion" by tomorrow 11:59pm (Nov 18). Let me know the topic and preferred date of presentation.

Last time:

If L is in NP, and $L \subset \Sigma^*$, Σ = alphabet of $\left\{ \begin{array}{l} \mathbb{L} \\ \text{Turing machine} \end{array} \right.$

then we had a way of:

give $w \in \Sigma^*$, $\rightsquigarrow$ Boolean formula, $f(w)$

s.t. ① $w \in L \iff f(w) \in SAT$

also $\left(\begin{array}{l} f \text{ is in 3CNF form} \\ f(w) \in 3SAT \end{array} \right)$

② $f(w)$ is of size poly in w

③ $f(w)$ can be produced in poly time by a Turing machine

(e.g. f involved x_{ijk}, $i = 0, \ldots, N$, $j = 1, \ldots, N+1$
$k = 1, \ldots, |\Gamma|$, $N = \text{poly}(|w|)$)

Defs: If L, L' languages, over alphabets Σ, Σ',

then $L \leq_p L'$ "L is reducible to L' in polynomial time" if there is a Turing machine, M, that on input w writes

$f(w)$, $f: \Sigma^* \to \Sigma'^*$, s.t.

① $w \in L \iff f(w) \in L'$ for all $w \in \Sigma^*$

② M halts/runs in time $p(|w|)$ for some fixed polynomial p.

We showed: $L \in NP$ then $L \le_p SAT$

and $L \le_p 3SAT$.

A function $f: \Sigma^* \to \Sigma'^*$ is poly time computable if the is a Turing machine that on input $w \in \Sigma^*$ writes $f(w)$ on one of its tapes and halts in poly time.

E.g., $\Sigma = \{1,2,3\}$, $\Sigma' = \{a\}$

$f : \{1,2,3\}^* \longrightarrow \{a\}^*$

Def: We say that L is NP-complete if

① $L \in NP$, ② if $\tilde{L} \in NP$, $\tilde{L} \leq_p L$.

e.g., SAT, 3SAT are NP-complete.

Today: SUBSET-SUM is NP-complete, so is VERTEX-EXPANSION, VERTEX COVER, ..., 3COLOUR, ...

So $NP = P$ iff $3SAT \in P$.

Also if L is NP-complete, then $NP = P \iff L \in P$.

Now: to show BLAH is NP-complete:

enough to show ① BLAH $\in NP$, ② $\underbrace{3SAT}_{\text{or any other NP-complete language}} \leq_p$ BLAH.

SUBSET-SUM (an "arithmetic" problem)

$$= \left\{ \langle\langle x_1, \ldots, x_m, t \rangle\rangle \;\middle|\; \begin{array}{l} m \in \mathbb{N},\ x_1, \ldots, x_m, t \in \mathbb{N} \\ \text{and for some} \\ I \subset [m], \\ \sum_{i \in I} x_i = t \end{array} \right\}$$

Idea! You have jobs that take time $x_1, \ldots, x_m$ and a computer/bin/... that has time t to run

Related problems in "load balancing" "bin packing" ...

Claim! Take $f \in 3CNF$, $f = f(x_1, \ldots, x_\ell)$ and is written as

C_1 AND C_2 AND ... AND C_k, each C_i OR of 3 literals

Take $F = (x_1 \text{ OR } x_2 \text{ OR } \neg x_3)$ AND $(x_2 \text{ OR } \neg x_1 \text{ OR } x_3)$

reduce F to a subset sum problem.

Idea! F is satisfiable iff for some

$x_1 = T, F$

$x_2 = T, F$

$\vdots$

$x_\ell = T, F$

"gadget"

each C_i is T

base 10

a digit

numbers

							C_1	C_2	
if $x_1=T$	y_1	1	0	0		0	1	0	
if $x_1=F$	z_1	1	0	0	-	0	0	1	
if $x_2=T$	y_2	0	1	0	-	0	1	1	
if $x_2=F$	z_2	0	1	0	-	0	0	0	
	y_3	$\vdots$	0	1		0 $\vdots$	0	1	
	z_3			1			1	0	
	$\vdots$			$\vdots$			$\vdots$	$\vdots$	
	y_ℓ	0	0	0		1	0	0	
	z_ℓ	0	0	0		1	0	0	
Dummy variables for clause 1	g_1	0	0	0	0	0	1	← 1 OR	0
	h_1	0	0	0	0	0	1	← 2	0
	g_2, h_2	0	0	0	...	0	0	1	
	$\vdots$	0	0	0	...	0	0	1	
target →	t	1	1	1	...	1	3 (4)	3	...

So one of y_1, z_1 will be chosen

" " y_2, z_2 " " " "

$\text{clause}_1 = x_1 \text{ OR } x_2 \text{ OR } \neg x_3$

take ~~z_1~~ y_1, y_2, z_3 (at least one)

Break 10:28–10:33

$(x_1 \text{ OR } x_2 \text{ OR } \neg x_3) \text{ AND } (x_2 \text{ OR } \neg x_1 \text{ OR } x_3)$

	x_1	x_2	x_3	c_1	c_2
y_1	1	0	0	1	0
z_1	1	0	0	0	1
y_2	0	1	0	1	1
z_2	0	1	0	0	0
y_3	0	0	1	0	1
z_3	0	0	1	1	0
g_1	0	0	0	1	0
h_1	0	0	0	1	0
g_2	0	0	0	0	1
h_2	0	0	0	0	1
t	1	1	1	3	3

1. build something in SUBSET-SUM represents $x_1 = T$ or $x_1 = F$, $x_2 = T$ or F, ..., $x_\ell = T/F$

2. build something in SUBSET-SUM to check if $C_1, C_2, \ldots, C_k$ are all satisfied

$\rightsquigarrow \{00010, 10001, 01011, 01000, \ldots, 00001, 00001, 11133\}$

$x_1 \; \cdots \; x_\ell \quad c_1 \; \cdots \; c_k$

y_1

z_1

$\vdots$

y_ℓ

z_ℓ

g_1

h_1

$\vdots$

g_k

h_k

t

digits $\ell + k$

of numbers $2\ell + 2k + 1$

Size problem: $(\ell+k)(2\ell+2k+1)$

3CNF has ℓ vars, k clauses

Next time!

VERTEX - EXPANSION

is NP-complete ...

Class ends

$L = \{ \langle M \rangle \mid M \text{ accepts at least one of its inputs} \}$

imagine you have a U = universal TM

$\langle M \rangle$ is over $\Sigma = \{1, 2, \ldots |\Sigma|\}$

say, $\Sigma = \{1, 2\}$

poss inputs: ε, 1, 2, 11, 12, 21, 22, ...

step 1

step 2

q_{acc}