CPSC 421/501 Nov 22

Today:

[We know: $3SAT \in P \iff P = NP$]

- 3SAT, SAT are "NP-complete"
- Other NP-complete languages

The rest of the course:

1. How to (maybe) show $P \neq NP$
2. How not to solve P vs. NP

[Ch 8, 9 Sip]

[Ch. 9 Sip]

Last time:

We took $L \in NP$, $L \subset \Sigma^*$,

then given $\sigma_1 \dots \sigma_n \in \Sigma^n$,

produced

$f(\sigma_1 \dots \sigma_n) \in$ 3CNF Boolean formula

s.t.

① $\sigma_1 \dots \sigma_n \in L$

$\iff f(\sigma_1 \dots \sigma_n)$ is satisfiable

Recall $f(\sigma_1, \dots, \sigma_n)$ had variables

$X_{ij\gamma}$, Y_{ij}, Z_{iq}

$i, j \in \{1, \dots, Cn^k\}$, $\gamma \in \Gamma$, $q \in Q$

From this, we can produce another 3CNF formula, with variables

$$x_1, \ldots, x_{p(n)}$$

$p(n)$ = polynomial in n.

$$3SAT = \left\{ \langle f \rangle \;\middle|\; \begin{array}{l} f \text{ is a 3CNF} \\ \text{Bool. form.} \\ \text{that is} \\ \text{satisfiable} \end{array} \right\}$$

e.g.

$$\langle X_{1000} \wedge X_2 \rangle$$

$$= X1000 \wedge X2$$

$$\in \{0, \ldots, 9, X, \wedge, \vee, \neg, (,)\}^*$$

From this:

$$3SAT \subset \Sigma_{SAT}^*$$

$$\Sigma_{SAT} = \{0, \ldots, 9, x, \wedge, \vee, \neg, (,)\}$$

$L \subset \Sigma^*$, $L \in NP$, Σ anything

We found

$$g : \Sigma^* \longrightarrow \Sigma_{SAT}^*$$

$$\sigma_1 \ldots \sigma_n \longmapsto \langle f(\sigma_1 \ldots \sigma_n) \rangle \in \Sigma_{SAT}^*$$

1. $\sigma_1 \ldots \sigma_n \in L \iff g(\sigma_1 \ldots \sigma_n)$

$$= \langle f(\sigma_1, \ldots, \sigma_n) \rangle$$

$$\in 3SAT$$

Reduced L to 3SAT ...

<u>Definition</u> Given $A \subset \Sigma_1^*$, $B \subset \Sigma_2^*$, we say that A is poly time reducible to B, written $A \leq_{\substack{\text{poly}\\ \text{time}}} B$ or $A \leq_p B$ if there is a poly time algorithm, i.e. a Turing machine, M, such that on input a string, $w \in \Sigma_1^*$, M computes a function

$$f : \Sigma_1^* \to \Sigma_2^*$$

s.t.

$$w \in A \iff f(w) \in B$$

and M runs in time $\leq$ poly(n), where $n = |w|$.

Cook-Levin Thm: If $L \in NP$,

then $L \leq_p 3SAT$

or $L \leq_p SAT$

Rem: $L \subset \Sigma_1^*$, Σ_1^* - decidable in time 1

But $A \leq_p B$,

So

$$A \le_{\text{blah}} B$$

means: a decider for B and can be used to decide A after running a computation of the form <u>blah</u>

$$A \le_{\substack{\text{poly}\\\text{time}}} B$$

Also, if $B \in P \Rightarrow A \in P$.

We've shown:

SAT, 3SAT are "NP-complete"

where we say

Def B is NP-complete, if

1. $B \in NP$
2. If $A \in NP$, then $A \leq_p B$.

In particular

$$A \in P \implies NP = P$$

$$A \notin P \implies NP \neq P$$

Not only are

SAT, 3SAT NP-complete

but

SUBSET-SUM, bin packing of some type

graph problems of some type, ...

=

SUBSET-SUM =

$$\left\{ \langle n_1, n_2, \ldots, n_m, t \rangle \;\middle|\; \text{for some } I \subset \{1, \ldots, m\} \text{ we have } \sum_{i \in I} n_i = t \right\}$$

e.g.

$$4, 5, 6, 7, 15 \;;$$

$$4 + 5 + 6 = 15$$

so $\langle 4, 5, 6, 7, 15 \rangle \in$ SUBSET-SUM

$\langle 1, 1, 1, 1, 100 \rangle \notin$ SUBSET-SUM

$$\langle 4, 5, 6, 7, 15 \rangle$$

$$= 4\#5\#6\#7\#15$$

$$\in \{0, \ldots, 9, \#\}^*$$

Also

$7\#6\#5\#4\#15 \in$ S-S

① SUBSET-SUM $\in$ NP

(don't forget this part)

Given

4 # 5 # 6 # 7 # 15

keep 4 ignore 4

keep 5 discard 5 . .

.

.

after going thru all but the last integer, take the sum, see if = last integer (here 15)

So "non-deterministically" "guess"

which of

$4, 5, 6, 7$

to keep, and sum what you've

kept.

Step ②: Given $L \in NP$,

is $L \le_p$ SUBSET-SUM

?

Let's show

3SAT $\le$ SUBSET-SUM

Since

$A \le_p B$, $B \le_p C$ then $A \le_p C$

Remark: If A is NP-complete

and $A \leq_p B$,

and $B \in NP$,

then B NP-complete.

3CNF

$$\langle (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \rangle$$

$\in$ 3SAT

iff a certain "SUBSET-SUM problem" lies in SUBSET-SUM