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 $L \in NP$, $L \subseteq \Sigma^*$, then given $\sigma_1 \ldots \sigma_n \in \Sigma^n$, produced

$$f(\sigma_1 \ldots \sigma_n) \in 3\text{CNF Boolean formula}$$

such that

1. $\sigma_1 \ldots \sigma_n \in L$  

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

Recall $f(\sigma_1 \ldots \sigma_n)$ had variables $X_{ijr}$, $Y_{ij}$, $Z_{iq}$

where $i, j \in \{1, \ldots, Cn^k\}$, $r \in \Gamma$, $q \in Q$
From this, we can produce another 3CNF formula, with variables

\[ x_1, \ldots, x_{p(n)} \]

\[ p(n) = \text{polynomial in } n. \]

\[ 3 \text{SAT} = \left\{ \langle f \rangle \mid f \text{ is a 3CNF} \right\} \]

\[ \text{Bool form, that is satifiable} \]

E.g.,

\[ \langle X_{1000} \land X_2 \rangle \]

\[ = X_{1000} \land X_2 \]

\[ \in \{0, 1, 9, x, \land, \lor, -, (, )\}^* \]
From this:

$$3\text{SAT} \leq^* \Sigma_{\text{sat}}$$

$$\Sigma_{\text{sat}} = \{0,1,9,x,\land,\lor,\neg,(),\}$$

$$L \in \Sigma^*, L \in \text{NP, } \exists \text{ anything}$$

We found

$$g : \Sigma^* \rightarrow \Sigma^*_{\text{sat}}$$

$$\sigma_1...\sigma_n \rightarrow \langle f(\sigma_1...\sigma_n) \rangle \in \Sigma^*_{\text{sat}}$$

1. $$\sigma_1...\sigma_n \in L \Rightarrow g(\sigma_1...\sigma_n) = \langle f(\sigma_1...\sigma_n) \rangle \in 3\text{SAT}$$
Reduced L to 3 SAT.

Definition: Given \( A \in \Sigma_1^* \), \( B \in \Sigma_2^* \), we say that

\[ A \text{ is poly time reducible to } B, \]

written \( A \leq_{\text{poly}} B \) or \( A \leq_{\text{p}} B \),

if there is a poly time algorithm, i.e., a Turing machine, \( M \), such that

on input a string, \( \omega \in \Sigma_1^* \),

\( M \) computes a function

\[ f : \Sigma_1^* \rightarrow \Sigma_2^* \]
s.t.,

\[ w \in A \Rightarrow f(w) \in B \]

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

=  

Cook-Levin Thm: If \( L \in \text{NP} \), then

\[ L \leq_p \text{3SAT} \]

or

\[ L \leq_p \text{SAT} \]

Rem: \( L \in \sum_1^p \), \( \sum_1^p \) - decidable in time \( l \)

But \( A \leq_p B \).
So

\[ A \leq B \]

means: a decider for \( B \) and can be used to decide \( A \) after running a computation of the form \( \text{blah} \)

\[ A \leq_{\text{poly time}} B \]

Also, if \( B \leq P = A \in P \).
We've shown:

SAT, 3SAT are \( \text{NP-complete} \)

where we say

\[
\text{Def } B \text{ is } \text{NP-complete}, \text{ if} \\
\begin{align*}
(1) & \quad B \in \text{NP} \\
(2) & \quad \text{If } A \in \text{NP}, \text{ then } A \leq_p B.
\end{align*}
\]

In particular

\[
\begin{align*}
A \in \text{P} & \Rightarrow \text{NP} = \text{P} \\
A \notin \text{P} & \Rightarrow \text{NP} \neq \text{P}
\end{align*}
\]
Not only are SAT, 3SAT NP-complete but subset-sum, bin packing of some type graph problems

\[ \text{SUBSET-SUM} = \left\{ \left< n_1, n_2, \ldots, n_m, t \right> \mid \text{for some } i \in \{1, \ldots, m\} \text{ we have } \sum_{i \in I} n_i = t \right\} \]
e.g. \[4, 5, 6, 7, 15;\]

\[4 + 5 + 6 = 15\]

So

\[\langle 4, 5, 6, 7, 15 \rangle \in \text{SUBSET-SUM}\]

\[\langle 1, 1, 1, 1, 100 \rangle \notin \text{SUBSET-SUM}\]

\[\langle 4, 5, 6, 7, 15 \rangle = 4 \# 5 \# 6 \# 7 \# 15\]

\[\in \{0, 1, \ldots, 9, \#\}^k\]

Also

\[\langle 7 \# 6 \# 5 \# 4 \# 15 \rangle \in S-S\]
1. \texttt{SUBSET-SUM} \in \text{NP} \\
(don’t forget this part)

Given \(\{4, 5, 6, 7, 15\}\)

\(\checkmark\) \(\checkmark\) \(\checkmark\) \(\checkmark\) \(\checkmark\)

keep 4 \(\checkmark\) ignore 4

keep 5 \(\checkmark\) discard \(\checkmark\)

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 \( L \leq_p \text{SUBSET-SUM} \).

Let's show
\( \text{3SAT} \leq \text{SUBSET-SUM} \).

Since
\( A \leq_p B, B \leq_p C \) then \( A \leq_p C \)
Remark: If \( A \) is NP-complete and \( A \leq_p B \), and \( B \in \text{NP} \), then \( B \) NP-complete.

3 CNF

\[
(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)
\]

\( \in \text{3SAT} \)

iff a certain "SUBSET-SUM problem" lies in \( \text{SUBSET-SUM} \)