Recap from last class:

For problems $A \& B$

$$A \rightarrow B \text{ or } A \leq B$$

means we can reduce $A$ to $B$:
- given a poly-time algorithm for $B$
  we can construct a poly-time algorithm for $A$.

SAT:

*input*: Boolean formula $f$ in CNF with
  $n$ variables $x_1, \ldots, x_n$

*output*: assignment to the variables that satisfies $f$
  if one exists
  & NO otherwise

What’s CNF?

CNF = conjunctive normal form

Variables $x_1, \ldots, x_n$ take values TRUE or FALSE

Literals $x_i, \overline{x_i}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}$

$\lor = OR, \land = AND$

Clauses are OR of several literals: example $(\overline{x_2} \lor x_3 \lor \overline{x_5})$

$f$ is AND of clauses:
example: $(\overline{x_3}) \land (\overline{x_2} \lor x_3 \lor \overline{x_5}) \lor (x_1 \lor x_5)$
Theorem: SAT is NP-complete

This means that:

a) SAT ∈ NP
b) for all $A \in$ NP, $A \rightarrow$ SAT

So unlikely to be a poly-time alg for SAT as that would imply a poly-time alg.
for every problem in NP.

3SAT:

**input:** Boolean formula $f$ in CNF with $n$ variables & $m$ clauses
where each clause has $\leq 3$ literals

**output:** Satisfying assignment if one exists & NO otherwise.

We'll show 3SAT is NP-complete.

Need to show:

a) 3SAT ∈ NP
b) SAT $\rightarrow$ 3SAT.
For (a): \(3\text{SAT} \in \text{NP}:\)

Given an assignment in \(O(1)\) time per clause, we can check that at least 1 literal is satisfied in every clause. Thus \(O(m)\) total time to check that \(f\) is satisfied.

For (b): \(\text{SAT} \rightarrow 3\text{SAT}.\)

Take input \(f\) for \(\text{SAT}.

We need to create input \(f'\) for \(3\text{SAT}.

Then given a satisfying assignment for \(f'\) we need to define a satisfying assignment for \(f\).

We also need to show that:

\[ f \text{ is satisfiable} \iff f' \text{ is satisfiable}. \]
Example: \( f = (x_3) \land (\overline{x_2} \lor x_3 \lor \overline{x_1} \lor \overline{x_4}) \land (x_2 \lor x_1) \)

Clauses \( C_1 \) & \( C_2 \) can stay the same but \( C_2 \) is too big for 3SAT.

Create a new variable \( y \).

Look at:
\[
C_2' = (\overline{x_2} \lor x_3 \lor y) \land (y \lor \overline{x_1} \lor \overline{x_4})
\]

Claim: \( C_2 \) is satisfiable \( \iff \) \( C_2' \) is satisfiable.

Proof:  

(\( \Rightarrow \)): Take a satisfying assignment to \( x_2, x_3, x_1, x_4 \) for \( C_2 \). This satisfies at least one of the clauses in \( C_2' \) & use \( y \) to satisfy the other.

(\( \Leftarrow \)): Take sat. assig. to \( C_2' \).
- if \( y = T \) then \( x_1 = F \) & \( x_4 = F \)
- if \( y = F \) then \( x_2 = F \) or \( x_3 = T \)
In either case, \( x_2 \) or \( x_3 \) or \( \overline{x_1} \) or \( x_4 \) is sat, so \( C_2 \) is satisfied.
Suppose \( C = (x_2 \lor x_3 \lor \overline{x}_1 \lor x_4 \lor x_5) \)

then create 2 new variables \( y_1 \& y_2 \)

Let \( C' = (x_2 \lor x_3 \lor y_1) \land (y_1 \lor x_1 \lor y_2) \land (x_2 \lor x_4 \lor x_5) \)

Claim: fix an assignment to \( x_1, \ldots, x_5 \)

then

\[ C \text{ is satisfied} \iff \text{there is an assignment to } y_1, y_2 \text{ that satisfies so that } C' \text{ is satisfied.} \]

In general, for

\[ C = (a_1 \lor a_2 \lor \ldots \lor a_k) \]

for literals \( a_1, \ldots, a_k \)

add \( k-3 \) new variables \( y_1, \ldots, y_{k-3} \)

and replace \( C \) by \( k-2 \) clauses:

\[ C' = (a_1 \lor a_2 \lor y_1) \land (y_1 \lor a_3 \lor y_2) \land (y_2 \lor a_4 \lor y_3) \land \ldots \land (y_{k-4} \lor a_{k-2} \lor y_{k-3}) \land (y_{k-3} \lor a_{k-1} \lor a_k) \]
Claim: for any assignment to $a_1, \ldots, a_k$

$C$ is satisfied $\iff$ There is an assignment to $y_1, \ldots, y_{k-2}$ so that $C'$ is satisfied.

Proof:

$(\Rightarrow)$ Take assignment to $a_1, \ldots, a_i$ satisfying $C$.

Let $a_i$ be min $i$ where $a_i$ is satisfied.

So $a_i = T \Rightarrow$ clause $i-1$ in $C'$ is satisfied.

Set $y_1 = y_2 = \ldots = y_{i-2} = T$

$\Rightarrow$ 1st $i-2$ clauses in $C'$ are satisfied.

Set $y_{i-1} = \ldots = y_k = F$

$\Rightarrow$ clauses $i, \ldots, k-2$ in $C'$ are satisfied.

$(\Leftarrow)$ Take assignment to $a_1, \ldots, a_k, y_1, \ldots, y_{k-2}$ sat. $C'$.

At least one of $\& a_i = T$, thus $C$ is satisfied.

Why?
Suppose $a_1 = a_2 = \ldots = a_k = F$

Then since $C'$ is satisfied

for clause 1 we must have $Y_1 = T$

for clause 2 $Y_2 = T$

\cdots

for clause $k-3$ we must have $Y_{k-3} = T$

and then clause $(\overline{Y_{k-3}} \lor \overline{v_{k-1}} \lor \overline{v_k}) = F$

So we have a contradiction.

---

**SAT $\rightarrow$ 3SAT**

Given $f$ for SAT

Create a new formula $f'$ by the following procedure:

For each clause $C$ in $f$,

if $C$ has $\leq 3$ literals keep it the same

if $C$ has $> 3$ literals then add $k-3$ new variables & replace $C$ by $C'$ as described before.
Use \( f' \) as input for 3SAT.

\( f \) is satisfiable \( \iff f' \) is satisfiable.

(\( \Rightarrow \)) Given assignment to \( x_1, \ldots, x_n \) satisfying \( f' \) for each \( C' \), there is an assignment to the \( k-3 \) new variables so that \( C' \) is satisfied.

(\( \Leftarrow \)) Given a satisfying assignment to \( f' \), for each \( C' \) in \( f' \) at least one of the literals in \( C \) is satisfied.

Given a satisfying assignment for \( f' \), ignore the new variables, and the assignment for \( x_1, \ldots, x_n \) is also a satisfying assignment for \( f \).