NP-COMPLETENESS:
THE COOK-LEVIN THEOREM

TUESDAY March 25
Theorem (Cook-Levin): SAT is NP-complete

Corollary: SAT ∈ P if and only if P = NP
Theorem (Cook/Levin'71) \( P = NP \iff \text{SAT} \in P \)
A 3cnf-formula is of the form:

\[(x_1 \lor \neg x_2 \lor x_3) \land (x_4 \lor x_2 \lor x_5) \land (x_3 \lor \neg x_2 \lor \neg x_1)\]
SAT = \{ \phi \mid \phi \text{ is a satisfiable boolean-formula} \}

3-SAT = \{ \phi \mid \phi \text{ is a satisfiable 3cnf-formula} \}

SAT, 3-SAT \in NP (why?)
Theorem (Cook-Levin): SAT is NP-complete

Proof:

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

(2) Every language \text{A} in \text{NP} is polynomial time reducible to \text{SAT}
Theorem (Cook-Levin): SAT is NP-complete

Proof:

(1) $\text{SAT} \in \text{NP}$ (3SAT $\in$ NP)

(2) Every language $A$ in NP is polynomial time reducible to SAT

We build a poly-time reduction from $A$ to SAT

The reduction turns a string $w$ into a 3-cnf formula $\phi$ such that $w \in A$ iff $\phi \in 3$-SAT. $\phi$ will simulate the NP machine $N$ for $A$ on $w$. 
Theorem (Cook-Levin): SAT is NP-complete

Proof:

(1) SAT ∈ NP (3SAT ∈ NP)

(2) Every language A in NP is polynomial time reducible to SAT

We build a poly-time reduction from A to SAT

The reduction turns a string w into a 3-cnf formula \( \phi \) such that \( w \in A \) iff \( \phi \in 3\text{-SAT} \).

\( \phi \) will simulate the NP machine N for A on w.

Let N be a non-deterministic TM that decides A in time \( n^k \)
The reduction $f$ turns a string $w$ into a 3-cnf formula $\phi$ such that: $w \in A \iff \phi \in 3SAT$. $\phi$ will simulate the NP machine $N$ for $A$ on $w$. 
So proof will also show:

3-SAT is NP-Complete
Deterministic
Computation

accept or reject

Non-Deterministic
Computation

\( n^k \)

\( \exp(n^k) \)

accept

reject
Suppose \( A \in \text{NTIME}(n^k) \) and let \( N \) be an NP machine for \( A \). A **tableau for \( N \) on \( w \)** is an \( n^k \times n^k \) table whose rows are the configurations of some possible computation of \( N \) on input \( w \).

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>q_0</th>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>\ldots</th>
<th>( w_n )</th>
<th>( \square )</th>
<th>( \ldots )</th>
<th>( \square )</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td></td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A tableau is **accepting** if any row of the tableau is an accepting configuration.

Determining whether $N$ accepts $w$ is equivalent to determining whether there is an accepting tableau for $N$ on $w$. 
A tableau is **accepting** if any row of the tableau is an accepting configuration.

Determining whether $N$ accepts $w$ is equivalent to determining whether there is an accepting tableau for $N$ on $w$.

Given $w$, our 3cnf-formula $\phi$ will describe a **generic** tableau for $N$ on $w$ (in fact, essentially **generic** for $N$ on any string $w$ of length $n$).

The 3cnf formula $\phi$ will be satisfiable *if and only if* there is an accepting tableau for $N$ on $w$. 
VARIABLES of $\phi$

Let $C = Q \cup \Gamma \cup \{\#\}$

Each of the $(n^k)^2$ entries of a tableau is a cell

$\text{cell}[i,j] = \text{the cell at row } i \text{ and column } j$

For each $i$ and $j$ ($1 \leq i, j \leq n^k$) and for each $s \in C$ we have a variable $x_{i,j,s}$

# variables $= |C|n^{2k}$, ie $O(n^{2k})$, since $|C|$ only depends on $N$
VARIABLES of \( \phi \)

Let \( C = Q \cup \Gamma \cup \{ \# \} \)

Each of the \((n^k)^2\) entries of a tableau is a cell

\[ \text{cell}[i,j] = \text{the cell at row } i \text{ and column } j \]

For each \( i \) and \( j \) (\( 1 \leq i, j \leq n^k \)) and for each \( s \in C \) we have a variable \( x_{i,j,s} \)

\# variables = \(|C|n^{2k}\), ie \( O(n^{2k}) \), since \(|C|\) only depends on \( N \)

These are the variables of \( \phi \) and represent the contents of the cells

We will have: \( x_{i,j,s} = 1 \iff \text{cell}[i,j] = s \)
\( x_{i,j,s} = 1 \)

means

\( \text{cell}[i,j] = s \)
We now design $\phi$ so that a satisfying assignment to the variables $x_{i,j,s}$ corresponds to an accepting tableau for $N$ on $w$.

The formula $\phi$ will be the AND of four parts:

$$\phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}}$$
We now design $\phi$ so that a satisfying assignment to the variables $x_{i,j,s}$ corresponds to an accepting tableau for $N$ on $w$

The formula $\phi$ will be the AND of four parts:

$$\phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}}$$

- $\phi_{\text{cell}}$ ensures that for each $i,j$, exactly one $x_{i,j,s} = 1$
- $\phi_{\text{start}}$ ensures that the first row of the table is the starting (initial) configuration of $N$ on $w$
- $\phi_{\text{accept}}$ ensures* that an accepting configuration occurs somewhere in the table
- $\phi_{\text{move}}$ ensures* that every row is a configuration that legally follows from the previous config

*if the other components of $\phi$ hold
\( \phi_{\text{cell}} \) ensures that for each \( i,j \), exactly one \( x_{i,j,s} = 1 \)

\[
\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[ \left( \bigvee_{s \in C} x_{i,j,s} \right) \land \left( \bigwedge_{s,t \in C \atop s \neq t} (\neg x_{i,j,s} \lor \neg x_{i,j,t}) \right) \right]
\]

- at least one variable is turned on
- at most one variable is turned on
\[ \phi_{\text{cell}} \] ensures that for each i,j, exactly one \( x_{i,j,s} = 1 \)

\[
\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[ \left( \bigvee_{s \in C} x_{i,j,s} \right)^{\bigwedge_{s,t \in C}} \left( \bigvee_{s \neq t} x_{i,j,s} \vee \neg x_{i,j,t} \right) \right]
\]

- at least one variable is turned on
- at most one variable is turned on

Thus, \( \phi_{\text{cell}} \) is satisfiable (ie, there exist assignment to the variables s.t. \( \phi_{\text{cell}} \) evaluates to 1)

\[ \iff \]

each cell in the tableau has exactly one symbol (from C.)
$\phi_{\text{start}} =$ 

$X_{1,1,#} \land X_{1,2,q_0} \land$

$X_{1,3,w_1} \land X_{1,4,w_2} \land \ldots \land X_{1,n+2,w_n} \land$

$X_{1,n+3,\square} \land \ldots \land X_{1,n^k-1,\square} \land X_{1,n^k,#}$
Thus, $\phi_{\text{start}}$ is satisfiable $\iff$ the first row of the tableau represents the start configuration for $N$ on input $w$.
Thus, $\phi_{\text{accept}}$ is satisfiable $\iff$ at least one cell in the tableau has the symbol $q_{\text{accept}}$. 

\[ \phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{accept}}} \]
\( \phi_{\text{move}} \) ensures that every row is a configuration that legally follows from the previous.

It works by ensuring that each \( 2 \times 3 \) “window” of cells is legal (Does not violate N’s rules).
\( \phi \text{move} \) ensures that every row is a configuration that legally follows from the previous.

It works by ensuring that each 2 \( \times \) 3 “window” of cells is legal (Does not violate N’s rules).

<table>
<thead>
<tr>
<th>#</th>
<th>( q_0 )</th>
<th>( w_1 )</th>
<th>( w_2 )</th>
<th>( \ldots )</th>
<th>( w_n )</th>
<th>( \square )</th>
<th>( \ldots )</th>
<th>( \square )</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>
If \( \delta(q_1,a) = \{(q_1,b,R)\} \) and \( \delta(q_1,b) = \{(q_2,c,L), (q_2,a,R)\} \), which of the following windows are legal:
If $\delta(q_1,a) = \{(q_1,b,R)\}$ and $\delta(q_1,b) = \{(q_2,c,L), (q_2,a,R)\}$, which of the following windows are legal:
CLAIM:
If
  • the top row of the table is the start configuration, and
  • and every window is legal,
Then
each row of the table is a configuration that legally follows the preceding one.
CLAIM:
If
• the top row of the table is the start configuration, and
• and every window is legal,

Then
each row of the table is a configuration that legally follows the preceding one.

Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.
CLAIM:
If
• the top row of the table is the start configuration, and
• and every window is legal,

Then
each row of the table is a configuration that legally follows the preceding one.

Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol.
CLAIM:
If
• the top row of the table is the start configuration, and
• and every window is legal,
Then
each row of the table is a configuration that legally follows the preceding one.

Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
**Proof:**

In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
**Proof:**

In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
Proof:
In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

Case 1. center cell of window is a non-state symbol and not adjacent to a state symbol
Case 2. center cell of window is a state symbol
**Proof:**

In upper configuration, every cell that doesn’t contain the boundary symbol #, is the center top cell of a window.

So the lower configuration follows from the upper!!!
The (i,j) Window

<table>
<thead>
<tr>
<th></th>
<th>col. j-1</th>
<th>col. j</th>
<th>col. j+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>row i</td>
<td>(i,j-1)</td>
<td>(i,j)</td>
<td>(i,j+1)</td>
</tr>
<tr>
<td></td>
<td>a_1</td>
<td>a_2</td>
<td>a_3</td>
</tr>
<tr>
<td>row i+1</td>
<td>(i+1,j-1)</td>
<td>(i+1,j)</td>
<td>(i+1,j+1)</td>
</tr>
<tr>
<td></td>
<td>a_4</td>
<td>a_5</td>
<td>a_6</td>
</tr>
</tbody>
</table>
$$\phi_{\text{move}} = \bigwedge \text{ (the (i, j) window is legal)} \quad 1 \leq i, j \leq n^k$$

the (i, j) window is legal =

$$\bigvee_{a_1, \ldots, a_6} \left( x_{i,j-1,a_1} \land x_{i,j,a_2} \land x_{i,j+1,a_3} \land x_{i+1,j-1,a_4} \land x_{i+1,j,a_5} \land x_{i+1,j+1,a_6} \right)$$

is a legal window

This is a disjunct over all ($\leq |C|^6$) legal sequences ($a_1, \ldots, a_6$).
\( \phi_{\text{move}} = \bigwedge \) (the \((i, j)\) window is legal)

\[ 1 \leq i, j \leq n^k \]

the \((i, j)\) window is legal =

\[ \bigvee_{a_1, \ldots, a_6} \left( x_{i,j-1,a_1} \land x_{i,j,a_2} \land x_{i,j+1,a_3} \land x_{i+1,j-1,a_4} \land x_{i+1,j,a_5} \land x_{i+1,j+1,a_6} \right) \]

is a legal window

This is a disjunct over all \((\leq |C|^6)\) legal sequences \((a_1, \ldots, a_6)\).

This disjunct is satisfiable

\( \iff \)

There is \textit{some} assignment to the cells (ie variables) in the window \((i,j)\) that makes the window legal
\[ \phi_{\text{move}} = \bigwedge \quad ( \text{the } (i, j) \text{ window is legal} ) \quad 1 \leq i, j \leq n^k \]

the \((i, j)\) window is legal =

\[ \bigvee \quad (x_{i,j-1,a_1} \land x_{i,j,a_2} \land x_{i,j+1,a_3} \land x_{i+1,j-1,a_4} \land x_{i+1,j,a_5} \land x_{i+1,j+1,a_6}) \quad a_1, \ldots, a_6 \]

is a legal window

This is a disjunct over all \((\leq |C|^6)\) legal sequences \((a_1, \ldots, a_6)\).

So \(\phi_{\text{move}}\) is satisfiable

\[ \Leftrightarrow \]

There is \textit{some} assignment to each of the variables that makes \textit{every} window legal.
\( \phi_{\text{move}} = \bigwedge ( \text{the (i, j) window is legal} ) \)

\[ 1 \leq i, j \leq n^k \]

the (i, j) window is legal =

\[ \bigvee_{a_1, \ldots, a_6} ( x_{i,j-1,a_1} \land x_{i,j,a_2} \land x_{i,j+1,a_3} \land x_{i+1,j-1,a_4} \land x_{i+1,j,a_5} \land x_{i+1,j+1,a_6} ) \]

is a legal window

This is a disjunct over all \( (\leq |C|^6) \) legal sequences \((a_1, \ldots, a_6)\).

Can re-write as equivalent conjunct:
\[ \phi_{\text{move}} = \bigwedge ( \text{the (i, j) window is legal} ) \]

1 \leq i, j \leq n^k

the (i, j) window is legal =

\[ \bigvee_{a_1, \ldots, a_6} (x_{i,j-1,a_1} \land x_{i,j,a_2} \land x_{i,j+1,a_3} \land x_{i+1,j-1,a_4} \land x_{i+1,j,a_5} \land x_{i+1,j+1,a_6} ) \]

is a legal window

This is a disjunct over all (\leq |C|^6) legal sequences \((a_1, \ldots, a_6)\).

Can re-write as equivalent conjunct:

\[ \equiv \bigwedge_{a_1, \ldots, a_6} (\bar{x}_{i,j-1,a_1} \lor \bar{x}_{i,j,a_2} \lor \bar{x}_{i,j+1,a_3} \lor \bar{x}_{i+1,j-1,a_4} \lor \bar{x}_{i+1,j,a_5} \lor \bar{x}_{i+1,j+1,a_6} ) \]

ISN'T a legal window

This is a conjunct over all (\leq |C|^6) illegal sequences \((a_1, \ldots, a_6)\).
$\phi = \phi_{cell} \land \phi_{start} \land \phi_{accept} \land \phi_{move}$

$\phi$ is satisfiable (ie, **there is some** assignment to each of the variables s.t. $\phi$ evaluates to 1)

$\iff$

**there is some** assignment to each of the variables s.t. $\phi_{cell}$ and $\phi_{start}$ and $\phi_{accept}$ and $\phi_{move}$ each evaluates to 1

$\iff$

**There is some** assignment of symbols to cells in the tableau such that:

- The first row of the tableau is a **start configuration** and
- Every row of the tableau is a configuration that follows from the preceding by the rules of $N$ and
- One row is an **accepting configuration**

$\iff$

**There is some** accepting computation for $N$ with input $w$
\[ \phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}} \]

**WHAT’S THE LENGTH OF \( \phi \) ?**
\[ \phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}} \]

\[ \phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[ \bigvee_{s \in C} x_{i,j,s} \land \left( \bigwedge_{s,t \in C \atop s \neq t} (\neg x_{i,j,s} \lor \neg x_{i,j,t}) \right) \right] \]

**O(n^{2k}) clauses**

\[ \text{Length}(\phi_{\text{cell}}) = O(n^{2k}) \cdot O(\log(n)) = O(n^{2k} \log n) \]

\[ \text{length}(indices) \]

\[ \时尚 (indices) \]
$$\phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}}$$

$$\phi_{\text{start}} = x_{1,1,#} \land x_{1,2,q_0} \land$$

$$x_{1,3,w_1} \land x_{1,4,w_2} \land \ldots \land x_{1,n+2,w_n} \land$$

$$x_{1,n+3,1} \land \ldots \land x_{1,rk-1,1} \land x_{1,rk,#}$$

$$O(n^k)$$
\[ \phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{accept}} \land \phi_{\text{move}} \]

\[ \phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j,q_{\text{accept}}} \]

\[ O(n^{2k}) \]
\[ \phi_{\text{move}} = \bigwedge ( \text{the (i, j) window is legal} ) \]
\[ 1 \leq i, j \leq n^k \]

the (i, j) window is legal =

\[ \bigwedge \left( \overline{x}_{i,j-1,a_1} \lor \overline{x}_{i,j,a_2} \lor \overline{x}_{i,j+1,a_3} \lor \overline{x}_{i+1,j-1,a_4} \lor \overline{x}_{i+1,j,a_5} \lor \overline{x}_{i+1,j+1,a_6} \right) \]

ISN’T a legal window

This is a conjunct over all (\( \leq |C|^6 \)) illegal sequences \((a_1, \ldots, a_6)\).

\[ O(n^{2k}) \]
Theorem (Cook-Levin): SAT is NP-complete

Corollary: SAT ∈ P if and only if P = NP
Theorem (Cook-Levin): 3SAT is NP-complete

Corollary: $3\text{SAT} \in P$ if and only if $P = NP$
3-SAT?

How do we convert the whole thing into a 3-cnf formula?

Everything was an AND of ORs
We just need to make those ORs with 3 literals

If a clause has less than three variables:

\[ a \equiv (a \lor a \lor a), \quad (a \lor b) \equiv (a \lor b \lor b) \]
3-SAT?

How do we convert the whole thing into a 3-cnf formula?

Everything was an AND of ORs
We just need to make those ORs with 3 literals

If a clause has less than three variables:
\[ a \equiv (a \lor a \lor a), \quad (a \lor b) \equiv (a \lor b \lor b) \]

If a clause has more than three variables:
\[ (a \lor b \lor c \lor d) \equiv (a \lor b \lor z) \land (\neg z \lor c \lor d) \]
\[ (a_1 \lor a_2 \lor \ldots \lor a_t) \equiv (a_1 \lor a_2 \lor z_1) \land (\neg z_1 \lor a_3 \lor z_2) \land \ldots \land (\neg z_{t-3} \lor a_{t-1} \lor z_t) \]
Given $A$ in NP. The reduction $f$ turned a string $w$ into a 3-cnf formula $\phi$ such that: $w \in A \iff \phi \in 3\text{SAT}$.
The reduction \( f \) is poly time. WHY?
3-SAT is NP-Complete
Theorem (Cook-Levin): 3SAT is NP-complete

Corollary: 3SAT ∈ P if and only if P = NP