THE COOK-LEVIN THEOREM
(SAT IS NP-COMPLETE)

Haoyue Ping
SAT

- Satisfiability Problem

- Given a Boolean formula, determine if there exists an interpretation that satisfies it.

- For example, given \( \phi = (\bar{x} \land y) \lor (x \land z) \)
  - This formula is satisfiable because it’s true when x is false, y is true and z is false.
NP-Complete

- A decision problem is NP-complete when it is both in NP and NP-hard.

- NP means
  - A nondeterministic Turing machine can solve in P
  - It’s verifiable in polynomial time by a deterministic Turing machine

- NP-hard means
  - "at least as hard as the hardest problems in NP"
  - Every problem in NP can be reduced to NP-hard problems in polynomial time
SAT is NP-Complete

- We need to prove that SAT is in both NP and NP-hard

- SAT is in NP
  - A nondeterministic Turing machine can “guess” an assignment and accept if this assignment satisfies $\phi$
  - Or given an assignment, a deterministic Turing machine can verify this assignment in polynomial time.

- SAT is NP-hard
  - Any language $A$ in NP is polynomial time reducible to SAT.
  - The IDEA is to turn Turing machine configurations into a SAT formula
Language $A$ in NP and its NTM decider

- Let $N$ be a nondeterministic Turing machine that decides $A$ in $n^k$ time for some constant $k$.
- Build a tableau to show the configurations and computation branches of $N$

  - Each row is a configuration of a branch of $N$
  - $N$ is in NP so
    - It stops in $n^k$ steps
    - Its header moves at most to column $n^k$
    - There are $n^{2k}$ cells in total
  - $N$ accept if there is an accepting configuration
Encoding the tableau into SAT formula

- Alphabet $C = Q \cup \Gamma \cup \{\#\}$
  - $Q$ is state set and $\Gamma$ is alphabet of $N$

- A variable $x_{i,j,s}$
  - When $x_{i,j,s} = 1$, a symbol $s$ from $C$ is in row $i$ and column $j$ of the tableau
  - When $x_{i,j,s} = 0$, otherwise

- Now design $\phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{move}} \land \phi_{\text{accept}}$
  - $\phi$ is satisfiable iff there exists an accepting configuration in tableau
  - $\phi$ is satisfiable iff NTM accepts $w$ in polynomial time.
\[ \phi = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{move}} \land \phi_{\text{accept}} \]

- \( \phi_{\text{cell}} \)
  - Every cell in tableau is legal, which means there is only one symbol in one cell.

- \( \phi_{\text{start}} \)
  - The first configuration is the starting configuration of \( N \) on input \( w \).

- \( \phi_{\text{move}} \)
  - The configurations are legal according to \( N \)'s transition function.

- \( \phi_{\text{accept}} \)
  - An accepting configuration occurs in tableau.
\( \phi_{\text{cell}} \)

- Every cell in tableau is legal

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

(At least one symbol is true) AND (There is no two different symbols to be true)
\( \phi_{\text{start}} \)

- The first configuration is the starting configuration of \( N \) on input \( w \).

\[
\phi_{\text{start}} = x_{1,1,\#} \land x_{1,2,\text{q}_0} \land x_{1,3,w_1} \land x_{1,4,w_2} \land \cdots \land x_{1,n+2,w_n} \land x_{1,n+3,\text{\textsc{\#}}} \land \cdots \land x_{1,n^k-1,\text{\textsc{\#}}} \land x_{1,n^k,\#}. 
\]
\( \phi_{\text{move}} \)

- The configurations are legal according to \( N \)'s transition function.

\[
\phi_{\text{move}} = \bigwedge_{1 \leq i < n^k, \ 1 < j < n^k} \text{(the \((i, j)\)-window is legal)}
\]

Every “window” is legal…
Window

- Window is of size $2 \times 3$
- Window verifies if lower 3 cells are legal, given upper 3 cells

IDEA
- Turing machine header only changes one cell in one move.
- The tape is of length $n^k$, so $n^k - 3$ cells stay the same.
Window Example

- When
  - $\delta(q_1, a) = \{(q_1, b, R)\}$
  - $\delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>q_1</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>q_2</td>
<td>a</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>q_1</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b)</td>
<td>a</td>
<td>a</td>
<td>q_2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>a</th>
<th>q_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c)</td>
<td>a</td>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>

Legal

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>q_1</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b)</td>
<td>q_2</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>b</th>
<th>q_1</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c)</td>
<td>q_2</td>
<td>b</td>
<td>q_2</td>
</tr>
</tbody>
</table>

Illegal
\( \phi \text{move} \)

- The configurations are legal according to \( N \)'s transition function.

\[
\phi_{\text{move}} = \bigwedge_{1 \leq i < n^k, 1 < j < n^k} \text{(the (i, j)-window is legal)}
\]

Every “window” is legal

Written formally, a window is legal when all six symbols satisfy the transition function

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

is a legal window
A configuration occurs in tableau.

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

The \( i \)th configuration is accepting.
Reduction

○ Any language $A \in NP$ is decided by a NTM $N$.

○ Execution of $N$ is denoted by configurations starting from reading input $w$.

○ These configurations form a tableau.

○ This tableau is of size $n^k \times n^k = n^{2k}$. Each cell contains one variable.

○ Encode this tableau into SAT formula, $\phi = \phi_{cell} \land \phi_{start} \land \phi_{move} \land \phi_{accept}$

○ $\phi$ will be of length $O(n^{2k} \log n)$. $\log n$ is to encode variable into binary integers.

○ $\phi$ guarantees that
  ○ $N$ starts with legal input.
  ○ $N$ moves legally
  ○ $N$ reaches accept iff $\phi$ is satisfiable.

○ **CONCLUSION** Since SAT is also in NP, SAT now is proved to be NP-complete.
Thanks!