Cook-Levin Theorem: The satisfiability problem of boolean expression is NP-complete.

Part 1: \( L_{\text{SAT}} \in \text{NP} \); easy.

Part 2: For any \( L \in \text{NP} \), \( L \leq_{\text{poly}} L_{\text{SAT}} \).

\[ \Rightarrow \text{For a nondet } M \text{ s.t. } L(M) = L \]

Given any \( w \), we have to construct a Turing computable polynomial-time function \( f \) s.t. \( f(w) \in L_{\text{SAT}} \) iff \( w \in L \).

Note: \( f(w) \) is a boolean expression (not too long).
ND TM \( M \) takes at most \( p(n) \) steps for any input of length \( n \) where \( p \) is some polynomial.

Final ID must be one of the final accepting states

\( S_M \) is a transition function that may have multiple successors (non-det)
\( w \in L \) if

1. \( M \) contains \( w \) as an initial input and
2. \( I_{p(n)} \) contains a final state
3. \( \forall j \leq p(n) I_{j+1} \) follows from \( I_{j} \) using a legal transition function of \( M \).

The goal is to represent the above conditions as a boolean formula \( F_{M}(w) \) which is satisfiable if \( w \in L \)

Reduction function \( f \) is computable in poly-time

\[ \Rightarrow |F_{M}(w)| \text{ is of polynomial length} \]

\[ \{1, 2, \ldots, k\} \]

Idea: To guess each of the \( p(n) \times p(n) \) symbols and verify conditions 1, 2, 3.

To convert from a \( k \)-valued variable to a boolean variable, we can introduce \( k \) variables for each variable.
$x_{i,1}, x_{i,2}, \ldots, x_{i,K}$, $x_{i,j}, x_{i,k}$

$X_1, X_2, \ldots, X_K$

1, 2, ..., $K$

$x_{i,j}$ are boolean variables st.

$x_{i,j} = \begin{cases} \text{true} & \text{if } x_i = j \in \{1, \ldots, K\} \\ \text{false} & \text{otherwise} \end{cases}$

Additional condition (beyond 1, 2, 3)

(4) Exactly one $x_{i,j}$ must be true for each $i \leq p(n) \times p(n)$

\[
(\overline{x_{i,1}} \lor x_{i,2} \lor \cdots \lor x_{i,K}) \land \left( \bigwedge_{j \neq j'} (x_{i,j} \Rightarrow \overline{x_{i,j'}}) \right)
\]

\[
(\overline{x_{i,j}} \lor \overline{x_{i,j'}})
\]

Total boolean variables: $p(n) \times p(n) \times K$