24.4
Proof of Cook-Levin Theorem
24.4.1

Statement and sketch of idea for the proof
Cook-Levin Theorem

**Theorem 24.1 (Cook-Levin).**

*SAT* is NP-Complete.

We have already seen that *SAT* is in *NP*.

Need to prove that every language \( L \in \text{NP}, L \leq_p \text{SAT} \)

**Difficulty:** Infinite number of languages in NP. Must simultaneously show a generic reduction strategy.
Cook-Levin Theorem

**Theorem 24.1 (Cook-Levin).**

*SAT is NP-Complete.*

We have already seen that *SAT* is in *NP*.

Need to prove that every language $L \in \text{NP}$, $L \leq_p \text{SAT}$

**Difficulty:** Infinite number of languages in *NP*. Must simultaneously show a generic reduction strategy.
The plot against SAT
High-level plan to proving the Cook-Levin theorem

What does it mean that $L \in \text{NP}$?
$L \in \text{NP}$ implies that there is a non-deterministic TM $M$ and polynomial $p()$ such that

\[ L = \{ x \in \Sigma^* \mid M \text{ accepts } x \text{ in at most } p(|x|) \text{ steps} \} \]

Input: $M, x, p$.
Question: Does $M$ stops on input $x$ after $p(|x|)$ steps?

Describe a reduction $R$ that computes from $M, x, p$ a SAT formula $\varphi$.
- $R$ takes as input a string $x$ and outputs a SAT formula $\varphi$
- $R$ runs in time polynomial in $|x|, |M|$
- $x \in L$ if and only if $\varphi$ is satisfiable
The plot against SAT

High-level plan to proving the Cook-Levin theorem

What does it mean that $L \in \text{NP}$?

$L \in \text{NP}$ implies that there is a non-deterministic TM $M$ and polynomial $p()$ such that

$$L = \{x \in \Sigma^* \mid M \text{ accepts } x \text{ in at most } p(|x|) \text{ steps}\}$$

Input: $M, x, p$

Question: Does $M$ stops on input $x$ after $p(|x|)$ steps?

Describe a reduction $R$ that computes from $M, x, p$ a SAT formula $\varphi$.

- $R$ takes as input a string $x$ and outputs a SAT formula $\varphi$
- $R$ runs in time polynomial in $|x|, |M|$
- $x \in L$ if and only if $\varphi$ is satisfiable
The plot against SAT

High-level plan to proving the Cook-Levin theorem

What does it mean that $L \in \text{NP}$?

$L \in \text{NP}$ implies that there is a non-deterministic TM $M$ and polynomial $p()$ such that

$$L = \{x \in \Sigma^* \mid M \text{ accepts } x \text{ in at most } p(|x|) \text{ steps}\}$$

Input: $M, x, p$.

Question: Does $M$ stops on input $x$ after $p(|x|)$ steps?

Describe a reduction $R$ that computes from $M, x, p$ a SAT formula $\varphi$.

- $R$ takes as input a string $x$ and outputs a SAT formula $\varphi$
- $R$ runs in time polynomial in $|x|, |M|$
- $x \in L$ if and only if $\varphi$ is satisfiable
The plot against SAT continued

\[ \langle x, M, p \rangle \xrightarrow{R} \varphi \]

poly-time computable

\( \varphi \) is satisfiable if and only if \( x \in L \)

\( \varphi \) is satisfiable if and only if nondeterministic \( M \) accepts \( x \) in \( p(|x|) \) steps

BIG IDEA

- \( \varphi \) will express “\( M \) on input \( x \) accepts in \( p(|x|) \) steps”
- \( \varphi \) will encode a computation history of \( M \) on \( x \)

\( \varphi \): CNF formula s.t if we have a satisfying assignment to it \( \implies \) accepting computation of \( M \) on \( x \) down to the last details (where the head is, what transition is chosen, what the tape contents are, at each step, etc).
The plot against SAT continued

\[ \langle x, M, p \rangle \rightarrow R \rightarrow \varphi \]

poly-time computable

\[ \varphi \] is satisfiable if and only if \( x \in L \)

\[ \varphi \] is satisfiable if and only if nondeterministic \( M \) accepts \( x \) in \( p(|x|) \) steps

BIG IDEA

- \( \varphi \) will express “\( M \) on input \( x \) accepts in \( p(|x|) \) steps”
- \( \varphi \) will encode a computation history of \( M \) on \( x \)

\( \varphi \): CNF formula s.t if we have a satisfying assignment to it \( \implies \) accepting computation of \( M \) on \( x \) down to the last details (where the head is, what transition is chosen, what the tape contents are, at each step, etc).
The plot against SAT continued

\[ \langle x, M, p \rangle \xrightarrow{R} \varphi \]

poly-time computable

\( \varphi \) is satisfiable if and only if \( x \in L \)
\( \varphi \) is satisfiable if and only if nondeterministic \( M \) accepts \( x \) in \( p(|x|) \) steps

**BIG IDEA**

- \( \varphi \) will express “\( M \) on input \( x \) accepts in \( p(|x|) \) steps”
- \( \varphi \) will encode a computation history of \( M \) on \( x \)

\( \varphi \): CNF formula s.t if we have a satisfying assignment to it \( \iff \) accepting computation of \( M \) on \( x \) down to the last details (where the head is, what transition is chosen, what the tape contents are, at each step, etc).
The plot against SAT continued

\[ \langle x, M, p \rangle \xrightarrow{R} \varphi \]

poly-time computable

\varphi \text{ is satisfiable if and only if } x \in L
\varphi \text{ is satisfiable if and only if nondeterministic } M \text{ accepts } x \text{ in } p(|x|) \text{ steps}

BIG IDEA

- \varphi \text{ will express } "M \text{ on input } x \text{ accepts in } p(|x|) \text{ steps}"
- \varphi \text{ will encode a computation history of } M \text{ on } x

\varphi: \text{ CNF formula s.t if we have a satisfying assignment to it } \implies \text{ accepting computation of } M \text{ on } x \text{ down to the last details (where the head is, what transition is chosen, what the tape contents are, at each step, etc).}
The Matrix Executions

Tableau of Computation

$M$ runs in time $p(|x|)$ on $x$. Entire computation of $M$ on $x$ can be represented by a “tableau”

Row $i$ gives contents of all cells at time $i$
At time 0 tape has input $x$ followed by blanks
Each row long enough to hold all cells $M$ might ever have scanned.
Four types of variables to describe computation of $M$ on $x$

- $T(b, h, i)$: tape cell at position $h$ holds symbol $b$ at time $i$.
  For $h = 1, \ldots, p(|x|)$, $b \in \Gamma$, $i = 0, \ldots, p(|x|)$.

- $H(h, i)$: read/write head is at position $h$ at time $i$.
  For $h = 1, \ldots, p(|x|)$, and $i = 0, \ldots, p(|x|)$.

- $S(q, i)$ state of $M$ is $q$ at time $i$.
  For all $q \in Q$ and $i = 0, \ldots, p(|x|)$.

- $I(j, i)$ instruction number $j$ is executed at time $i$.
  $M$ is non-deterministic, need to specify transitions in some way. Number transitions as $1, 2, \ldots, \ell$ where $j$th transition is $< q_j, b_j, q_j', b_j', d_j >$ indication $(q_j', b_j', d_j) \in \delta(q_j, b_j)$, direction $d_j \in \{-1, 0, 1\}$.

Number of variables is $O(p(|x|)^2|M|^2)$.
Notation

Some abbreviations for ease of notation

\(\bigwedge_{k=1}^m x_k\) means \(x_1 \land x_2 \land \ldots \land x_m\)

\(\bigvee_{k=1}^m x_k\) means \(x_1 \lor x_2 \lor \ldots \lor x_m\)

\(\bigoplus(x_1, x_2, \ldots, x_k)\) is a formula that means exactly one of \(x_1, x_2, \ldots, x_m\) is true. Can be converted to \text{CNF} \text{ form}

**CNF** formula showing making sure that at most one variable is assigned value \(1\):

\[
\bigwedge_{1 \leq i < j \leq k} (\overline{x}_i \lor \overline{x}_j)
\]

Making sure that one of the variables is true: \(\bigvee_{i=1}^{k} x_i\).

\[
\bigoplus(x_1, x_2, \ldots, x_k) = \bigwedge_{1 \leq i < j \leq k} (\overline{x}_i \lor \overline{x}_j) \land (x_1 \lor x_2 \lor \cdots \lor x_k).
\]
Notation

Some abbreviations for ease of notation
\[ \bigwedge_{k=1}^{m} x_k \] means \( x_1 \land x_2 \land \ldots \land x_m \)

\[ \bigvee_{k=1}^{m} x_k \] means \( x_1 \lor x_2 \lor \ldots \lor x_m \)

\( \bigoplus(x_1, x_2, \ldots, x_k) \) is a formula that means exactly one of \( x_1, x_2, \ldots, x_m \) is true. Can be converted to CNF form

CNF formula showing making sure that at most one variable is assigned value 1:

\[ \bigwedge_{1 \leq i < j \leq k} (\overline{x}_i \lor \overline{x}_j) \]

Making sure that one of the variables is true: \( \bigvee_{i=1}^{k} x_i \).

\[ \bigoplus(x_1, x_2, \ldots, x_k) = \bigwedge_{1 \leq i < j \leq k} (\overline{x}_i \lor \overline{x}_j) \bigwedge (x_1 \lor x_2 \lor \ldots \lor x_k) . \]
Notation

Some abbreviations for ease of notation

\( \bigwedge_{k=1}^{m} x_k \) means \( x_1 \land x_2 \land \ldots \land x_m \)

\( \bigvee_{k=1}^{m} x_k \) means \( x_1 \lor x_2 \lor \ldots \lor x_m \)

\( \bigoplus(x_1, x_2, \ldots, x_k) \) is a formula that means exactly one of \( x_1, x_2, \ldots, x_m \) is true. Can be converted to CNF form.

CNF formula showing making sure that at most one variable is assigned value 1:

\[
\bigwedge_{1 \leq i < j \leq k} (\overline{x_i} \lor \overline{x_j})
\]

Making sure that one of the variables is true: \( \bigvee_{i=1}^{k} x_i \).

\[
\bigoplus(x_1, x_2, \ldots, x_k) = \bigwedge_{1 \leq i < j \leq k} (\overline{x_i} \lor \overline{x_j}) \land (x_1 \lor x_2 \lor \cdots \lor x_k).
\]
Notation

Some abbreviations for ease of notation

\( \bigwedge_{k=1}^{m} x_k \) means \( x_1 \land x_2 \land \ldots \land x_m \)

\( \bigvee_{k=1}^{m} x_k \) means \( x_1 \lor x_2 \lor \ldots \lor x_m \)

\( \bigoplus(x_1, x_2, \ldots, x_k) \) is a formula that means exactly one of \( x_1, x_2, \ldots, x_m \) is true. Can be converted to CNF form

CNF formula showing making sure that at most one variable is assigned value 1:

\[
\bigwedge_{1\leq i<j\leq k} (\overline{x}_i \lor \overline{x}_j)
\]

Making sure that one of the variables is true: \( \bigvee_{i=1}^{k} x_i \).

\[
\bigoplus(x_1, x_2, \ldots, x_k) = \bigwedge_{1\leq i<j\leq k} (\overline{x}_i \lor \overline{x}_j) \land (x_1 \lor x_2 \lor \cdots \lor x_k).
\]
Clauses of $\varphi$

$\varphi$ is the conjunction of 8 clause groups:

$$\varphi = \bigwedge_{i=1}^{12} \varphi_i$$

where each $\varphi_i$ is a CNF formula. Described in subsequent slides.

**Property:** $\varphi$ is satisfied $\iff$ there is an execution of $M$ on $x$ that accepts the language in $p(|x|)$ time.
THE END

...

(for now)