The Cook-Levin Theorem

CS468/568

Author: David Costanzo, Yale University
Adapted from Computational Complexity: A Modern Approach by Arora and Barak
The Claim

• Claim 1: \textit{SAT} is NP-complete

• Claim 2: \textit{3SAT} is NP-complete
Proof Structure

- SAT \in NP
- SAT is NP-complete
- SAT is NP-hard
- 3SAT is NP-complete
- 3SAT is NP-hard
- 3SAT \in NP

Lemma 2.11
- SAT \leq_p 3SAT

Lemma 2.14
Lemma 2.14: $\text{SAT} \leq_p \text{3SAT}$

- Convert each CNF clause of size $k$ to equivalent clauses of sizes $k-1$ and 3
  - repeat until all clauses are size 3

\[
C = u_1 \lor u_2 \lor \cdots \lor u_k
\]
\[
C_1 = u_1 \lor u_2 \lor y \quad \text{(fresh variable)}
\]
\[
C_2 = u_3 \lor \cdots \lor u_k \lor \overline{y}
\]

$C \in \text{SAT} \iff C_1 \land C_2 \in \text{SAT}$
Lemma 2.11: \textbf{SAT} is NP-hard

• Pick any \( L \in \text{NP} \), and let \( M \) be a poly-time TM recognizing \( L \) on any input \( x \) and valid certificate \( u \)

• Assumptions:
  – \( M \) has 1 input tape and 1 work/output tape
  – \( M \) is oblivious

\[ x \in L \iff \exists u \in \{0,1\}^p(|x|) . M(x \cdot u) = 1 \]
Lemma 2.11: \textbf{SAT} is NP-hard

- Need to describe a poly-time function mapping input $x$ of length $n$ to formula $\varphi_x$ such that:

\[ x \in L \iff \varphi_x \in \text{SAT} \]
Lemma 2.11: \( \text{SAT} \) is NP-hard

- Need to describe a poly-time function mapping input \( x \) of length \( n \) to formula \( \varphi_x \) such that:
  \[
  x \in L \iff \varphi_x \in \text{SAT}
  \]

Or equivalently:

\[
\exists u \in \{0,1\}^{p(|x|)} . \ M(x \cdot u) = 1 \iff \varphi_x \in \text{SAT}
\]
Lemma 2.11: \textbf{SAT} is NP-hard

• Variables of $\varphi_x$:
  - $y_1, y_2, \ldots, y_n$ (first $n$ bits of input $x \cdot u$)
  - $y_{n+1}, y_{n+2}, \ldots, y_{n+p(n)}$ (next $p(n)$ bits of input $x \cdot u$)
  - $z_1, z_2, \ldots, z_{T(n)}$ (snapshots of TM $M$ on input $x \cdot u$)

(bit strings of length $c$)
Lemma 2.11: \textbf{SAT} is NP-hard

• Variables of $\varphi_x$:
  
  – $y_1, y_2, \ldots, y_n$ (first $n$ bits of input $x \cdot u$)
  
  – $y_{n+1}, y_{n+2}, \ldots, y_{n+p(n)}$ (next $p(n)$ bits of input $x \cdot u$)
  
  – $z_1, z_2, \ldots, z_{T(n)}$ (snapshots of TM $M$ on input $x \cdot u$)

• The goal:

  The formula $\varphi_x$ is satisfied iff the input snapshots represent a valid execution of $M$ on input $y = x \cdot u$
Lemma 2.11: SAT is NP-hard

• What is a snapshot?

input tape: \( \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \end{bmatrix} \ldots \)

work/output tape: \( \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix} \ldots \)

state register: \( q_7 \)

snapshot: \( (0, 1, q_7) \)

(constant # of bits)
Lemma 2.11: $\text{SAT}$ is NP-hard

• What is a snapshot?

input tape $\triangleright 0 \ 1 \ 0 \ 0 \ 1 \ \ldots$

work/output tape $\triangleright 1 \ 1 \ 1 \ 0 \ 0 \ \ldots$

state register $q_7$

snapshot $= (0, 1, q_7)$

(constant # of bits)

• How to determine the correct snapshot at time $i$?
Lemma 2.11: \textsc{Sat} is NP-hard

- Given input $y$ and all snapshots $z_1, \ldots, z_{i-1}$, there exists \textit{at most one} valid snapshot $z_i$
- This $z_i$ depends \textit{only} on $z_{i-1}$, $z_{\text{prev}(i)}$, and $y_{\text{inputpos}(i)}$
  - $\text{prev}(i) =$ the most recent step preceding $i$ at which the work/output head was at the same position as at step $i$
  - $\text{inputpos}(i) =$ the position of the input head at step $i$
  - These are well-defined by obliviousness, and poly-time computable by simulating $M$ on dummy input $0^{|x|}$

\[
    z_i = F(z_{i-1}, z_{\text{prev}(i)}, y_{\text{inputpos}(i)})
\]

$F: \{0,1\}^{2c+1} \rightarrow \{0,1\}^c$
Lemma 2.11: \textbf{SAT} is NP-hard

\[ \varphi_x = A \land B \land C \land D \]

- \textbf{A}: the first \( n \) bits of \( y \) are equal to the bits of \( x \)
  \[ (x_1 \lor \overline{y}_1) \land (x_1 \lor y_1) \land \cdots \land (x_n \lor \overline{y}_n) \land (x_n \lor y_n) \]
- \textbf{B}: \( z_1 \) correctly encodes the initial snapshot of \( M \)
- \textbf{C}: \( z_{T(n)} \) correctly encodes a halting snapshot of \( M \) from which \( M \) outputs 1 (i.e. accept)
- \textbf{D}: For each \( i \) between 2 and \( T(n) - 1 \),
  \[ z_i = F(z_{i-1}, z_{prev(i)}, y_{inputpos(i)}) \]
Lemma 2.11: SAT is NP-hard

How to express $z_i = F(z_{i-1}, z_{\text{prev}(i)}, y_{\text{inputpos}(i)})$ in CNF?

$f: \{0,1\}^l \rightarrow \{0,1\}$

$$\psi_x = \bigwedge_{v: f(v) = 0} C_v(x_1, \ldots, x_l)$$

$C_{1101} = \overline{x_1} \lor \overline{x_2} \lor x_3 \lor \overline{x_4}$

$C_{00110} = x_1 \lor x_2 \lor \overline{x_3} \lor \overline{x_4} \lor x_5$

eetc...

$$\psi_x \in SAT \iff f(x) = 1$$
QED