- NP Completeness (Section 7.4)

Some material from slides by M. Sipser
Announcements

- Homework 4 assigned today
  - due Wednesday March 24, 2021 at 11:59pm (NY Time)
  - Only 3 problems (a half homework)

- Exam 2 is Monday, March 29, 2021
A function \( f: \Sigma^* \rightarrow \Sigma^* \) is a polynomial time computable function if there exists a polynomial time Turing machine \( M \) that, on every input \( w \), halts with just \( f(w) \).

Language \( A \) is polynomial time mapping reducible (also called polynomially time reducible) to language \( B \) (written \( A \leq_p B \)) if there is a polynomial time computable function \( f \) where \( w \in A \) if and only if \( f(w) \in B \).

The function \( f \) is called a polynomial time reduction of \( A \) to \( B \).

**Theorem:** If \( A \leq_p B \) and \( B \in \mathbb{P} \) then \( A \in \mathbb{P} \)
A language \( B \) is **NP-complete** if it satisfies two conditions:

1. \( B \) is in NP and
2. Every \( A \) in NP is polynomial time reducible to \( B \)

If we can prove that any NP-complete problem has a polynomial time solution, then we have proved that \( P = NP \).

**Theorem:** If \( A \leq_p B \) and \( B \in P \) then \( A \in P \)
A **Boolean formula** $\phi$ has Boolean variables (True/False values) and the Boolean operations: And ($\land$), Or ($\lor$), and Not ($\neg$).

$\phi$ is **satisfiable** if $\phi$ evaluates to $\text{TRUE}$ for some assignment to its variables.

$SAT = \{\langle \phi \rangle | \phi$ is a satisfiable Boolean formula$\}$

**The Cook-Levin Theorem:** SAT is NP-complete.

- $SAT \in P \iff P = NP$
Theorem: SAT is NP-complete

1. SAT is in NP
2. Every $A$ in NP is polynomial time reducible to SAT

Proof of part 1:
Theorem: SAT is NP-complete

1. SAT is in NP
2. Every A in NP is polynomial time reducible to SAT

Proof outline for part 2:
A **tableau** for $M$ on $w$ is an $n^k \times n^k$ table, represents one computation path in $M$ on input $w$.

- Each row is a configuration: shows current state, current tape contents, current head location.
- A tableau is **accepting** if any row of the tableau is an accepting configuration.
Now we describe the function $f$ that constructs $\phi_{M,w}$

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

- Each cell in a tableau takes a value from $C$

Define variables of $\phi_{M,w}$ as $x_{i,j,s}$

- $x_{i,j,s} = 1$ means cell $(i,j)$ takes the value $s \in C$

$\phi_{M,w}$ consists of 4 parts:

$$\phi_{M,w} = \phi_{cell} \land \phi_{start} \land \phi_{move} \land \phi_{accept}$$
\[
\phi_{\text{cell}} = \bigwedge_{1 \leq i, j \leq n^k} \left[ \left( \bigvee_{s \in C} x_{i,j,s} \right) \wedge \left( \bigwedge_{s,t \in C} \left( \overline{x_{i,j,s}} \vee \overline{x_{i,j,t}} \right) \right) \right].
\]
\[ \phi_{\text{start}} = x_{1,1}, \# \land x_{1,2}, q_0 \land \\
x_{1,3}, w_1 \land x_{1,4}, w_2 \land \ldots \land x_{1,n+2}, w_n \land \\
x_{1,n+3}, \sqcup \land \ldots \land x_{1,n^k-1}, \sqcup \land x_{1,n^k}, \# \]

\[ \phi_{\text{accept}} = \bigvee_{1 \leq i, j \leq n^k} x_{i,j}, q_{\text{accept}} \]
\[ \phi_{M,w} = \phi_{\text{cell}} \land \phi_{\text{start}} \land \phi_{\text{move}} \land \phi_{\text{accept}} \]

**Legal windows:** consistent with M’s transition function

**Illegal windows:** not consistent with M’s transition function
\[ \phi_{\text{move}} = \bigwedge_{1<i \leq n^k, 1<j<n^k} (\text{the } (i,j) \text{ window is legal}) \]

“the \((i,j)\) window is legal” is formalized as:

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

is a legal window
We have given a reduction $f$ that computes $\phi_{M,w}$ such that $\phi_{M,w}$ has a satisfying assignment if and only if $M$ accepts $w$

$$\phi_{M,w} = \phi_{cell} \land \phi_{start} \land \phi_{move} \land \phi_{accept}$$

- Is $f$ a polynomial time reduction?
  - How many variables in $\phi_{M,w}$?
  - How long is $\phi_{M,w}$?
A language $B$ is **NP-complete** if it satisfies two conditions:

1. $B$ is in NP and
2. Every $A$ in NP is polynomial time reducible to $B$

**Theorem:** If $B$ is NP-complete and $B \leq_p C$ for $C$ in NP, then $C$ is NP-complete.
A Boolean formula $\phi$ is in **Conjunctive Normal Form** (CNF) if it has the form

$\phi = (x \lor \overline{y} \lor z) \land (\overline{x} \lor \overline{s} \lor z \lor u) \land \cdots \land (\overline{z} \lor \overline{u})$

A **3cnf-formula** is a cnf-formula where all clauses have exactly three literals.

$3SAT = \{ \langle \phi \rangle | \phi \text{ is a satisfiable 3cnf-formula}\}$

**Theorem:** $3SAT$ is NP-complete.
3SAT = \{⟨φ⟩| φ is a satisfiable 3cnf-formula\} \quad \quad SAT = \{⟨φ⟩| φ is a satisfiable Boolean formula\}

**Theorem:** 3SAT is NP-complete.