Last time:
- $\text{NTIME}(t(n))$, NP
- P vs NP problem
- Dynamic Programming, $A_{CFG} \in P$
- Polynomial-time reducibility

Today:
- NP-completeness
Quick Review

**Defn:** A is polynomial time reducible to B \((A \leq_p B)\) if \(A \leq_m B\) by a reduction function that is computable in polynomial time.

**Theorem:** If \(A \leq_p B\) and \(B \in P\) then \(A \in P\).

\[ f \text{ is computable in polynomial time} \]

NP = All languages where can verify membership quickly
P = All languages where can test membership quickly

**P versus NP question:** Does P = NP?

\[ SAT = \{\langle \phi \rangle | \phi \text{ is a satisfiable Boolean formula} \} \]

**Cook-Levin Theorem:** \(SAT \in P \rightarrow P = NP\)

**Proof plan:** Show that every \(A \in NP\) is polynomial time reducible to \(SAT\).
**Example:** \(3SAT\) and \(CLIQUE\)

**Defn:** A Boolean formula \(\phi\) is in **Conjunctive Normal Form (CNF)** if it has the form:

\[
\phi = (x \lor \bar{y} \lor z) \land (x \lor \bar{s} \lor z \lor u) \land \cdots \land (\bar{z} \lor \bar{u})
\]

- **Literal:** a variable or a negated variable
- **Clause:** an OR (\(\lor\)) of literals.
- **CNF:** an AND (\(\land\)) of clauses.
- **3CNF:** a CNF with exactly 3 literals in each clause.

\(3SAT = \{ \langle \phi \rangle \mid \phi \) is a satisfiable 3CNF formula\}

**Defn:** A \(k\)-clique in a graph is a subset of \(k\) nodes all directly connected by edges.

\(CLIQUE = \{ \langle G, k \rangle \mid \) graph \(G\) contains a \(k\)-clique\}

Will show: \(3SAT \leq_p CLIQUE\)
**Theorem:** $3SAT \leq_P CLIQUE$

**Proof:** Give polynomial-time reduction $f$ that maps $\phi$ to $G, k$ where $\phi$ is satisfiable iff $G$ has a $k$-clique.

A satisfying assignment to a CNF formula has $\geq 1$ true literal in each clause.

\[
\phi = (a \lor b \lor \overline{c}) \land (\overline{a} \lor b \lor d) \land (a \lor c \lor \overline{e}) \land \cdots \land (\overline{x} \lor y \lor \overline{z})
\]

\[\begin{align*}
\phi & \downarrow f \\
G & = \\
k & = \# \text{ clauses}
\end{align*}\]

Forbidden edges:
1) within a clause
2) inconsistent labels ($a$ and $\overline{a}$)

$G$ has all non-forbidden edges.
\[ 3\text{SAT} \leq_P \text{CLIQUE} \text{ conclusion} \]

\[ \phi = (a \lor b \lor \overline{c}) \land (\overline{a} \lor b \lor d) \land (a \lor c \lor \overline{e}) \land \cdots \land (\overline{x} \lor y \lor \overline{z}) \]

**Claim:** \( \phi \) is satisfiable iff \( G \) has a \( k \)-clique

(\( \rightarrow \)) Take any satisfying assignment to \( \phi \). Pick 1 true literal in each clause.

The corresponding nodes in \( G \) are a \( k \)-clique because they don’t have forbidden edges.

(\( \leftarrow \)) Take any \( k \)-clique in \( G \). It must have 1 node in each clause.

Set each corresponding literal \( \text{TRUE} \). That gives a satisfying assignment to \( \phi \).

The reduction \( f \) is computable in polynomial time.

**Corollary:** \( \text{CLIQUE} \in P \rightarrow 3\text{SAT} \in P \)

**Check-in 15.1**

Does this proof require 3 literals per clause?

(a) Yes, to prove the claim.

(b) Yes, to show it is in poly time.

(c) No, it works for any size clauses.
**Defn:** $B$ is **NP-complete** if

1) $B \in \text{NP}$
2) For all $A \in \text{NP}$, $A \leq_p B$

If $B$ is NP-complete and $B \in \text{P}$ then $\text{P} = \text{NP}$.

**Cook-Levin Theorem:** $\text{SAT}$ is NP-complete

**Proof:** Next lecture; assume true

---

**Check-in 15.2**

What language that we’ve previously seen is most analogous to $\text{SAT}$?

(a) $A_{TM}$

(b) $E_{TM}$

(c) $\{0^k1^k \mid k \geq 0\}$
**Theorem:** \( HAMPATH \) is NP-complete

Proof: Show \( 3SAT \leq_p HAMPATH \) (assumes \( 3SAT \) is NP-complete)

Idea: “Simulate” variables and clauses with “gadgets”

\[
\phi = (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4) \land \ldots \land (\quad)
\]

\[
f \downarrow
\]

\[
\langle G, s, t \rangle
\]

variable gadget

clause gadget

Zig-zag \( \iff \) Corresponds to setting \( x_1 \) TRUE

Zag-zig \( \iff \) Corresponds to setting \( x_1 \) FALSE
Construction of $G$

$$\phi = (x_1 \lor \overline{x_2} \lor x_3) \land (x_1 \lor x_2 \lor x_4) \land \cdots \land (x_m)$$

Claim: $\phi$ is satisfiable if and only if $G$ has a Hamiltonian path from $s$ to $t$.

→ Take any satisfying assignment to $\phi$.

Make corresponding zig-zags and zag-zigs through variable gadgets from $s$ to $t$.

Make detours to visit the clause nodes $c_i$.

← Take any Hamiltonian path from $s$ to $t$.

Show it must be zig-zags and zag-zigs with detours to visit all $c_i$.

Get corresponding truth assignment. It must satisfy $\phi$ because path visits all $c_i$.

The reduction $f$ is computable in polynomial time.

Check-in 15.3

Would this construction still work if we made $G$ undirected by changing all the arrows to lines? In other words, would this construction show that the undirected Hamiltonian path problem is NP-complete?

(a) Yes, the construction would still work.

(b) No, the construction depends on $G$ being directed.
Quick review of today

1. NP-completeness
2. $SAT$ and $3SAT$
3. $3SAT \leq_p HAMPATH$
4. $3SAT \leq_p CLIQUE$
5. Strategy for proving NP-completeness:
   Reduce from $3SAT$ by constructing gadgets that simulate variables and clauses.