Review

- **P** – class of problems with polytime algorithm.
- **NP** – class of problems with polytime certifier.

<table>
<thead>
<tr>
<th>Problem (X)</th>
<th>Independent-Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance (s)</td>
<td>Graph G and number k</td>
</tr>
<tr>
<td>Algorithm (A)</td>
<td>No poly-time algorithm known</td>
</tr>
<tr>
<td>Hint (t)</td>
<td>Which nodes are in the answer?</td>
</tr>
<tr>
<td>Certifier (C)</td>
<td>Are those nodes independent and size k?</td>
</tr>
</tbody>
</table>

**Circuit-SAT**

**Problem**: Given a circuit built of AND, OR, and NOT gates with some hard-coded inputs, is there a way to set remaining inputs so the output is 1?

Satisfiable? Yes. Set inputs: 1, 1, 0.

**Cook-Levin Theorem**

*Circuit-SAT* is NP-Complete.

**Proof Idea**: encode arbitrary certifier \( C(s, t) \) as a circuit

- If \( X \in \text{NP} \), then \( X \) has a poly-time certifier \( C(s, t) \)
- \( s \) is Yes instance \( \iff \exists t \) such that \( C(s, t) \) outputs Yes
- \( s \) is Yes instance \( \iff \) circuit is satisfiable
- Algorithm for *Circuit-SAT* implies an algorithm for \( X \)
A Circuit-SAT reduction

- Vertex Cover – Does $G$ have VC of size at most $k$?

Proving New Problems NP-Complete

Suppose $X$ is in NP.

Fact: If $Y$ is NP-complete and $Y \leq_P X$, then $X$ is NP-complete.

Want to prove problem $X$ is NP-complete
- Check $X \in$ NP.
- Choose known NP-complete problem $Y$.
- Prove $Y \leq_P X$.

Clicker

It’s easy to show that $3$-$SAT \leq_P$ Circuit-SAT. What can we conclude from this?

A. $3$-$SAT$ is NP-complete.
B. $3$-$SAT$ is in NP.
C. If there is no polynomial time algorithm for $3$-$SAT$, then there is no polynomial time algorithm for Circuit-SAT.

Proving New Problems NP-Complete

Theorem: $3$-$SAT$ is NP-Complete.
1. In NP? Yes, check satisfying assignment in poly-time.
2. Prove by reduction from Circuit-SAT.

Example.

From Circuit-SAT to 3-SAT

Fact: If $Y$ is NP-complete, $X$ is in NP, and $Y \leq_P X$, then $X$ is NP-complete.

Theorem: 3-SAT is NP-Complete.
1. In NP? Yes, check satisfying assignment in poly-time.
2. Can show that Circuit-SAT $\leq_P$ 3-SAT

Reduction: Circuit-SAT $\leq_P$ 3-SAT
- One variable $x_v$ per circuit node $v$ plus clauses to enforce circuit computations
- Equality = equivalence (conjunction of two implications)
- Write implication $A \Rightarrow B$ as clause $\neg A \lor B$
- Negation node: $x_v = \neg x_w$
  - $x_u \Rightarrow \neg x_v$
  - $\neg x_u \Rightarrow x_v$
- AND node: $x_v = x_u \land x_w$
  - $x_v \Rightarrow x_u$
  - $x_v \Rightarrow x_w$
  - $\neg x_v \Rightarrow \neg x_u \lor \neg x_w$
- OR node: $x_v = x_u \lor x_w$
  - $x_u \Rightarrow x_v$
  - $x_w \Rightarrow x_v$
  - $x_v \Rightarrow x_u \lor x_w$
Reduction: Circuit-Sat \( \leq_p 3\text{-Sat} \)

- Clause \( C = x_v \) for input bits \( v \) fixed to one
- Clause \( C = \neg x_v \) for input bits \( v \) fixed to zero
- Clause \( C = x_o \) for output bit
- This formula is satisfiable iff circuit is satisfiable.
- Deal with clauses of size 1 and 2 by introducing two new variables and clauses that force them to be equal to zero.

Clicker Question

Which of the following statements is NOT true?

A. \( \text{SAT} \leq_p 3\text{-SAT} \)
B. \( 3\text{-SAT} \leq_p \text{SAT} \)
C. \( k\text{-SAT} \leq_p \text{SAT} \) for all \( k \geq 2 \)
D. \( k\text{-SAT} \) is NP-complete for all \( k \geq 2 \)

NP-Complete Problems So Far

Theorem: IndependentSet, VertexCover, SetCover, SAT, 3-SAT are all NP-Complete.

NP-Complete Problems: Preview

Traveling Salesman Problem

- TSP. Given \( n \) cities and distance function \( d(i, j) \), is there a tour that visits all cities with total distance less than \( D \)?
  - Tour: ordering of cities \( i_1, i_2, \ldots, i_n \) with \( i_1 = 1 \)
  - Distance is \( \sum_{j=1}^{n-1} d(i_j, i_{j+1}) + d(i_n, 1) \)

  Applications: traveling salesman, moving robotic arms

  Let’s prove a simpler problem is NP-complete, and then use it to show TSP is NP-complete.

Hamiltonian Cycle Problem

- \( \text{HamCycle} \) – Hamiltonian Cycle. Given directed graph \( G = (V, E) \), is there a cycle that visits each vertex exactly once?

  \[ \begin{align*}
  v_1 & \quad v_2 \quad v_3 \\
  v_4 & \quad v_5 & \quad v_6 \\
  \end{align*} \]

  \( v_1, v_3, v_2, v_5, v_4, v_6 \) is a Hamiltonian Cycle
Theorem. HAM-CYCLE is NP-Complete.

- It is in NP.
- Need to reduce from some NP-Complete problem. Which one?

Claim. 3-SAT ≤ₚ HAM-CYCLE.

Reduction has two main parts.

- Make a graph with 2ⁿ Hamiltonian cycles, one per assignment.
- Augment graph with clauses to invalidate assignments.

Reduction: Graph skeleton

\[ x_i = 1 \iff \text{traverse } P_i \text{ from } L \rightarrow R \]

Reduction: Clause Gadgets

\[ C_1 = x_1 \lor \overline{x}_2 \lor x_3 \]

Reduction: Details

- n rows (bidirected paths) \( P_1, \ldots, P_n \) (one per variable)
- Each row has 3m + 3 vertices, connected to neighbors in forward/backward direction
- First and last vertex of row i connected to first and last of i + 1.
- Source s connected to first and last of row 1.
- First and last of row n connected to t.
- Edge \((t, s)\)
- Skeleton has 2ⁿ possible Hamiltonian Cycles, corresponding to truth assignments to \( x_1, \ldots, x_n \)
  - Traverse \( P_i \) L to R \( \iff x_i = 1 \)
  - Traverse \( P_i \) R to L \( \iff x_i = 0 \)

Reduction: High-Level

- Correspondence between Hamiltonian cycles and truth assignments
  - \( x_i = 1 \): traverse path \( P_i \) from \( L \rightarrow R \)
  - \( x_i = 0 \): traverse path \( P_i \) from \( R \rightarrow L \)

- Node \( c_j \) for clause \( C_j \) must be visited in middle of some \( P_i \)
  - \( x_i \in C_j \): can visit \( c_j \) during \( L \rightarrow R \) traversal of \( P_i \).
  - \( x_i = 1 \) satisfies \( C_j \)
  - \( x_i \in C_j \): can visit \( c_j \) during \( R \rightarrow L \) traversal of \( P_i \).
  - \( x_i = 0 \) satisfies \( C_j \)

- There is a Hamiltonian cycle
  \( \iff \) can visit all clause nodes
  \( \iff \) there is a truth assignment that satisfies all clauses

Reduction: Clause Gadgets

For each clause \( C_l \) construct gadget to restrict possible truth assignments

- New node \( c_l \)
  - If \( x_i \in C_l \)
    - Add edges \((v_i, 3i, c_l)\) and \((c_l, v_i, 3i+1)\)
    - \( c_l \) can be visited during \( L \rightarrow R \) traversal of \( P_i \)
  - If \( \neg x_i \in C_l \)
    - Add edges \((v_i, 3i+1, c_l)\) and \((c_l, v_{i+1})\)
    - \( c_l \) can be visited during \( R \rightarrow L \) traversal of \( P_i \)
### Proof of Correctness

Given a satisfying assignment, construct Hamiltonian Cycle

- If $x_i = 1$ traverse $P_i$ from $L \rightarrow R$, else $R \rightarrow L$.
- Each $C^i$ is satisfied, so one path $P_i$ is traversed in the correct direction to “splice” $C^i$ into our cycle.
- The result is a Hamiltonian Cycle.

Given Hamiltonian cycle, construct satisfying assignment:

- If cycle visits $c_j$ from row $i$, it will also leave to row $i$ because of “buffer” nodes.
- Therefore, ignoring clause nodes, cycle traverses each row completely from $L \rightarrow R$ or $R \rightarrow L$.
- Set $x_i = 1$ if $P_i$ traversed $L \rightarrow R$, else $x_i = 0$.
- Every node $c_j$ visited $\Rightarrow$ every clause $C_j$ is satisfied.

### Traveling Salesman

TSP. Given $n$ cities and distance function $d(i, j)$, is there a tour that visits all cities with total distance less than $D$?

**Theorem.** TSP is NP-Complete

- Clearly in NP.
- Reduction? From Ham-Cycle

### Clicker

We want to show that Ham-Cycle $\leq_T$ TSP. How can we do so?

Given a HamCycle instance $G = (V, E)$ make TSP instance with one city per vertex and...

- A. $d(v_i, v_j) = 1$ if $(v_i, v_j) \in E$, else 2. Tour distance: $\leq n$?
- B. $d(v_i, v_j) = 2$ if $(v_i, v_j) \in E$, else 1. Tour distance: $\leq n$?
- C. $d(v_i, v_j) = 1$ if $(v_i, v_j) \in E$, else 2. Tour distance: $\leq m$?

### Ham-Path

Similar to Hamiltonian Cycle, visit every vertex exactly once.

**Theorem.** Ham-Path is NP-Complete.

Two proofs.

- Modify 3-SAT to Ham-Cycle reduction.
- Show that Ham-Cycle reduces to Ham-Path

### NP-Complete Problems

- Circuit-SAT
- 3-SAT
- 3-SAT
- Indep-Set
- Ham-Cycle
- Ham-Cycle
- Vertex-Cover
- Traveling-Salesman
- Traveling-Salesman
- Set-Cover
- Set-Cover