Recap

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

<table>
<thead>
<tr>
<th>Problem (X)</th>
<th>Instance (s)</th>
<th>Algorithm (A)</th>
<th>Hint (t)</th>
<th>Certifier (C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDENT-SET</td>
<td>Graph G and number k</td>
<td>Try all subsets and check (but not poly-time)</td>
<td>Which nodes are in the answer?</td>
<td>Are those nodes independent and size k?</td>
</tr>
</tbody>
</table>

Example

Proving New Problems NP-Complete

Claim: If Y is NP-complete and Y \( \leq_p \) X, then X is NP-complete.

Theorem: 3-SAT is NP-Complete.

- In NP? Yes, check satisfying assignment in polytime.
- Prove by reduction from CircuitSAT.

Example.

\[ \begin{align*}
1 & \quad \downarrow \\
2 & \quad 3 \\
Y & \quad x \\
\end{align*} \]

Discuss reduction on board.

Reduction: Circuit-Sat \( \leq_p \) 3-SAT

- One variable \( x_v \) per circuit node \( v \) plus clauses to enforce circuit computations
- For exposition, encode logical implications using "or" clauses
  - \( A \Rightarrow B \) is the same as \( \neg A \lor B \)
  - \( B \) can additionally be a disjunction
- Negation node: \( x_v = \neg x_u \)
  - \( x_u \Rightarrow \neg x_v \)
  - \( \neg x_u \Rightarrow x_v \)
- OR node: \( x_v = x_u \lor x_w \)
  - \( x_u \Rightarrow x_v \)
  - \( x_w \Rightarrow x_v \)
  - \( x_v \Rightarrow x_u \lor x_w \)
- AND node: \( x_v = x_u \land x_w \)
  - \( x_u \Rightarrow x_v \)
  - \( x_w \Rightarrow x_v \)
  - \( \neg x_v \Rightarrow \neg x_u \lor \neg x_w \)

NP-Complete Problems So Far

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

Want to prove problem $X$ is NP-complete
- Check $X \inNP$.
- Choose known NP-complete problem $Y$.
- Prove $Y \leq_P X$.
- Usually suffices to do single transformation $sY \rightarrow sX$ s.t.
  - $sX$ is Yes instance iff $sY$ is Yes instance

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 salesperson, 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

- Hamiltonian Cycle. Given directed graph $G = (V, E)$, is there a cycle that visits each vertex exactly once?
- $v_1, v_3, v_2, v_5, v_4, v_6$ is a Hamiltonian Cycle

HamCycle

Theorem. HamCycle is NP-Complete.
- It is in NP.
- Need to reduce from some NP-Complete problem. Which one?
Claim. 3-SAT $\leq_P$ HamCycle.
Reduction has two main parts.
- Make a graph with $2^n$ Hamiltonian cycles, one per assignment.
- Augment graph with clauses to invalidate assignments.
Board work.

Reduction: Graph skeleton

- $n$ rows (bidirected paths) $P_1, \ldots, P_n$ (one per variable)
- 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 $2^n$ 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: Clause Gadgets

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

- New node \( c_\ell \)
- If \( x_i \in C_\ell \)
  - Add edges \((v_i, 3\ell + 1, c_\ell)\) and \((c_\ell, v_i, 3\ell + 1)\)
- \( c_\ell \) can be visited during L to R traversal of \( P_i \)
- If \( \neg x_i \in C_\ell \)
  - Add edges \((v_i, 3\ell + 1, c_\ell)\) and \((c_\ell, v_i, 3\ell)\)
- \( c_\ell \) can be visited during R to 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 \( C_\ell \) is satisfied, so one path \( P_i \) is traversed in the correct direction to “splice” \( c_\ell \) into our cycle
- The result is a Hamiltonian Cycle

Given Hamiltonian cycle, construct satisfying assignment:

- If cycle visits \( c_\ell \) 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

HamPath

Similar to Hamiltonian Cycle, visit every vertex exactly once.

Theorem. HamPath is NP-Complete.

Two proofs.

- Modify 3-SAT to HamCycle reduction.
- Reduce from HamCycle directly.

NP-Complete Problems

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