Today’s Topics

- Formal Definition: P, NP
- Review: PATH
  - PATH $\in$ P
- Review: HAMPATH
  - HAMPATH $\in$ NP
- Polynomial Time Reducibility
- Example Reductions — Proving NP-Complete
  - $3$-SAT $\leq_p$ SUBSET-SUM
  - HAMPATH $\leq_p$ UHAMPATH
Formal Definition: $P$

TIME

- Let $t: \mathbb{N} \rightarrow \mathbb{N}$
- Say $M$ runs in time $t$ if TM $M$ always halts within $t(n)$ steps on all inputs of length $n$
Formal Definition: P (cont.)

\[ \text{TIME}(n^k) = \{ B \mid \text{some 1-tape deterministic TM decides } B \text{ in } O(t(n^k)) \text{ steps} \} \]

\[ P = \bigcup_k \text{TIME}(n^k) \]

= polynomial time decidable languages

Corresponds roughly to realistically solvable problems
Formal Definition: NP

NTIME

- Let $t : \mathbb{N} \to \mathbb{N}$
- A NTM $M$ runs in time $t(n)$ if all branches halt within $t(n)$ steps on all inputs of length $n$
- $\text{NTIME}(t(n)) = \{ B \mid \text{some 1-tape NTM decides } B \text{ and runs in } O(t(n)) \text{ steps} \}$
Formal Definition: NP (cont.)

\[ NP = \bigcup_k \text{NTIME}(n^k) \]

= nondeterministic polynomial time decidable languages

Corresponds roughly to easily verifiable problems
Intuitions: P vs. NP

P - All languages where one can **test** membership quickly
  - Problem presented to nondet. TM solvable in polynomial time

NP - All languages where one can **verify** membership quickly
  - Problem + solution presented to nondet. TM verified in polynomial time

P ⊆ NP, but unknown whether P = NP or P ≠ NP
Review: PATH

PATH = \{ <G, s, t> \mid G \text{ is a directed graph with path from } s \text{ to } t \} \]

Thm: PATH ∈ P

Proof: M = “On input <G, s, t>
1. Run BFS on G starting at s
2. Accept if t is reached.
   Reject otherwise”
Review: HAMPATH

HAMPATH = \{<G, s, t> \mid G \text{ is a directed graph with path from } s \text{ to } t \\
\quad \text{and path goes through every node of } G \text{ without repeats } \}

Thm: HAMPATH \in NP

Proof: M = “On input <G, s, t> (G has m nodes)

1. Nondeterministically pick sequence v_1, v_2, \ldots, v_m of m nodes
2. Accept if v_1 = s, v_m = t \\
   \quad \text{each } (v_i, v_{i+1}) \text{ is an edge and } v_i \text{ does not repeat}
3. Reject if any condition fails”
Polynomial Time Reducibility

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

Thm: If A \leq_p B and B \in P, then A \in P

\[ f \text{ is computable in polynomial time} \]
Polynomial Time Reducibility (cont.)

Corollary: If SAT ∈ P, then P = NP

Idea to show SAT ∈ P → P = NP
Define: SUBSET-SUM

Language definition

Given a collection of numbers $x_1, ..., x_k$ and a target number $t$

Does the collection contain a subcollection of numbers which sum up to $t$?

ex) $\{1, 2, 3, 5, 7\}, \ t = 13 \in \text{SUBSET-SUM}$

\[ t = 20 \notin \text{SUBSET-SUM} \]
Example Reduction: $3$-SAT $\leq_p$ SUBSET-SUM

Proving SUBSET-SUM is NP-Complete

1. SUBSET-SUM $\in$ NP
2. NP-Complete Language (3-SAT) $\leq_p$ SUBSET-SUM

Simpler Question First: Is SUBSET-SUM $\in$ NP?
Example Reduction: 3-SAT $\leq_p$ SUBSET-SUM (cont.)

Proving SUBSET-SUM is NP-Complete: 3-SAT $\leq_p$ SUBSET-SUM

Idea:

- Find way to convert *any* 3-SAT problem to a SUBSET-SUM problem
- SUBSET-SUM problem should somehow simulate solving 3-SAT formula
- Make sure conversion is polynomial in time!

Therefore, solving SUBSET-SUM problem, basically also solving 3-SAT problem
Example Reduction: $3\text{-SAT} \leq_p \text{SUBSET-SUM}$ (cont.)

Construction: Assume $l$ variable $x_1 \ldots x_l$, assume $k$ clauses $c_1 \ldots c_k$

For every variable $x_i$ produce two digits $y_i$, $z_i$ (for SUBSET-SUM problem)

For every clause $c_j$ produce two digits $g_j$, $h_j$ (for SUBSET-SUM problem)
Example Reduction: $3\text{-SAT} \leq_p \text{SUBSET-SUM}$ (cont.)

Construction:

$$(x_1 \lor \overline{x_2} \lor x_3) \land (x_2 \lor x_3 \lor \cdots) \land \cdots \land (\overline{x_3} \lor \cdots \lor \cdots)$$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>\ldots</th>
<th>l</th>
<th>c_1</th>
<th>c_2</th>
<th>\ldots</th>
<th>c_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>y_1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>z_1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>y_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>z_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>y_3</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>z_3</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>y_1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z_1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_1</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h_1</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_2</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h_2</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g_k</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h_k</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>\ldots</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Represents variables

Represents clauses
Define: UHAMPATH

Recall:

HAMPATH = \{<G, s, t> \mid G \text{ is a directed graph with path from } s \text{ to } t \\
and path goes through every node of } G \text{ without repeats } \}

UHAMPATH = \{<G, s, t> \mid G \text{ is a undirected graph with path from } s \text{ to } t \\
and path goes through every node of } G \text{ without repeats } \}
Example Reduction: $\text{HAMPATH} \leq_p \text{UHAMPATH}$ cont.

Proving UHAMPATH is NP-Complete

1. $\text{UHAMPATH} \in \text{NP}$
2. NP-Complete Language (HAMPATH) $\leq_p \text{UHAMPATH}$

Simpler Question First: Is UHAMPATH $\in$ NP?
Example Reduction: $\text{HAMPATH} \leq_p \text{UHAMPATH} \text{ cont.}$

Proving UHAMPATH is NP-Complete: $\text{HAMPATH} \leq_p \text{UHAMPATH}$

Idea:

- Convert HAMPATH directed graph $G$ to UHAMPATH undirected graph $G'$ where:
  - $<G, s, t> \in \text{HAMPATH}$ iff $<G', s', t'> \in \text{UHAMPATH}$
  - $<G, s, t> \notin \text{HAMPATH}$ iff $<G', s', t'> \notin \text{UHAMPATH}$

- Make sure conversion is polynomial in time!
Example Reduction: \( HAMPATH \leq_p UHAMPATH \) cont.

Construction: Convert every node \( u \) in HAMPATH \( G \), to three nodes in \( G' \)

- \( u \rightarrow u^{in}, u^{mid}, u^{out} \)
- \( s \rightarrow s^{out} \)
- \( t \rightarrow t^{in} \)

HAMPATH path: \( s, u_1, u_2, \ldots, u_k, t, \)

gets converted to

UHAMPATH path: \( s^{out}, u_1^{in}, u_1^{mid}, u_1^{out}, u_2^{in}, u_2^{mid}, u_2^{out}, \ldots, t^{in} \)
Example Reduction: $\text{HAMPATH} \leq_p \text{UHAMPATH cont.}$

Construction by example:
Example Reduction: $\text{HAMPATH} \leq_p \text{UHAMPATH cont.}$

Construction by example: