Recap

- Problem $X$ is a set of strings $s$, the YES instances.
- Algorithm solves $X$ if $A(s) = \text{true}$ iff $s \in X$.
- $B$ is polytime certifier for $X$ if
  - $B$ is polytime algorithm of two inputs $s$ and $t$ (a hint).
  - $s \in X$ iff exists $t$ with $|t| \leq p(|s|)$ and $B(s, t) = \text{True}$.
- $P$ – class of problems with polytime algorithm.
- $NP$ – class of problems with polytime certifier.

Example

- **Problem ($X$)**: INDEPENDENTSET
- **Instance ($s$)**: Graph $G$ and number $k$
- **Algorithm ($A$)**: Try all subsets and check
- **Hint ($t$)**: Which nodes are in the answer?
- **Certifier ($B$)**: Are those nodes independent and size $k$?

Plan for today

- Review $3\text{-SAT} \leq_P \text{CIRCUITSAT}$
- $\text{HAMCYCLE}$
- $\text{TSP}$

The Reduction

- One variable $x_v$ per circuit node $v$.
- Clauses to enforce circuit computations.
  - If $v$ is $\neg$ then $v$ has one input $u$ and can add clauses $(x_v \lor x_u), (\neg x_v \lor \neg x_u)$.
  - If $v$ is $\lor$ with $u, w$ incoming then
    $(x_v \lor \neg x_u), (x_v \lor \neg x_w), (\neg x_v \lor x_u \lor x_w)$.
  - If $v$ is $\land$ then
    $(\neg x_v \lor x_u), (\neg x_v \lor x_w), (x_v \lor \neg x_u \lor \neg x_w)$.
- Input bits get set with $(x_v)$ if fixed to one and $(\neg x_v)$ otherwise.
- Clause $(x_v)$ for output bit.

Back to 3-SAT

**Claim.** If $Y$ is NP-complete and $Y \leq_P X$, then $X$ is NP-complete.

**Theorem.** $3\text{-SAT}$ is NP-Complete.

- Clearly in $NP$.
- Prove by reduction from $\text{CIRCUITSAT}$.

Example.
Final steps

- This formula satisfiable iff circuit is satisfiable.
- But not a 3-sat formula! It has clauses of size 1 and 2.
  - Fix: 4 new variables $z_1, \ldots, z_4$ where $z_1, z_2$ forced to be 0.
  - Include those two in any short clause.

**Theorem.** \textsc{IndependentSet}, \textsc{VertexCover}, \textsc{SetCover}, \textsc{SAT}, 3-SAT are all NP-Complete.

Finding NP-Complete Problems.

Want to prove problem $X$ is NP-complete.
- Check $X \in \mathcal{NP}$.
- Choose known NP-complete problem $Y$.
- Prove $Y \leq_P X$.
- Often suffices to do single transformation from $y \to x$ where
  - $y \in Y$ if $x \in X$.
  - $y \notin Y$ if $x \notin X$.
  - Known as Karp Reduction.

Touring problems.

Two new problems.
- TSP – Traveling Salesman. Given points $v_1, \ldots, v_n$ with distances $d(v_i, v_j) \geq 0$, can we visit all points and return home with total distance less than $B$?

$$\text{cost}(\sigma) = \sum_{i=1}^{n} d(v_{\sigma(i)}, v_{\sigma(i+1)})$$

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

HamCycle Example

Theorem. \textsc{HamCycle} is NP-Complete.
- It is in $\mathcal{NP}$.
- Need to reduce from some NP-Complete problem. Which one?

Claim. 3-SAT $\leq_P \textsc{HamCycle}$.
Reduction has two main parts.
- Make a graph with $2^n$ Hamiltonian cycles, one per assignment.
- Augment graph with clauses to invalidate assignments.

Graph skeleton
### Skeleton Construction

- $n$ rows (one per variable).
- Row has $4k + 2$ vertices connected in forward and backward path.
- 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 $(t, s)$.

### Augmenting

For clause $C_ℓ = x_i ∨ ¬x_j ∨ x_k$ new node $c_ℓ$ in graph.
- Edges $(v_i, 4_ℓ, c_ℓ)$ and $(c_ℓ, v_i, 4_ℓ + 1)$.
- Edges $(v_j, 4_ℓ + 1, c_ℓ)$ and $(c_ℓ, v_j, 4_ℓ)$.
- Edges $(v_k, 4_ℓ, c_ℓ)$ and $(c_ℓ, v_k, 4_ℓ + 1)$.

Can only visit $c_ℓ$ on row $i$ if traverse $i$ from left to right.

### Example

$$(x_1 ∨ x_2 ∨ ¬x_3) \land (¬x_1 ∨ ¬x_2 ∨ ¬x_3)$$

### Proof

If $φ$ is satisfying assignment
- If $φ(x_i) = 1$ traverse left to right, else right to left.
- For each $C_ℓ$, it is satisfied, so one term is traversed in the correct direction
- We can therefore splice it into our cycle.

If $P$ is a Hamiltonian cycle
- If $P$ visits $c_ℓ$ from row $i$, it will also leave to row $i$.
- Splice out clause variables leaves cycle on skeleton.
- Cycles on skeleton correspond to assignments!

### Traveling Salesman

- TSP – Traveling Salesman. Given points $v_1, \ldots, v_n$ with distances $d(v_i, v_j) \geq 0$, can we visit all points and return home with total distance less than $B$?

$$\text{cost}(σ) = \sum_{i=1}^{n} d(v_σ(i), v_σ(i+1))$$

**Theorem.** TSP is NP-Complete
- Clearly in $NP$.
- Reduction from HamCycle.

### TSP reduction

Given HamCycle instance $G = (V, E)$ make TSP instance
- One point per vertex.
- $d(v_i, v_j) = 1$ if $(v_i, v_j) \in E$, else 2. (asymmetric).
- Set bound to be $n$.

TSP of distance $n$ iff HamCycle of length $n$.
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.

Graph Coloring

Def. A k-coloring of a graph $G = (V, E)$ is a function $f : V \rightarrow \{1, \ldots, k\}$ such that for all $(u, v) \in E$, $f(u) \neq f(v)$.

Problem. Given $G = (V, E)$ and number $k$, does $G$ have a $k$-coloring?

Many applications
- Actually coloring maps!
- Scheduling jobs on machine with competing resources.
- Allocating variables to registers in a compiler.

Claim. 2-coloring $\in P$.

Proof.
- 2-coloring equivalent to bipartite testing.
- From discussion section.

Theorem. 3-coloring is NP-Complete.