NP and Computational Intractability
Polynomial-Time Reduction

Desiderata’. Suppose we could solve X in polynomial-time. What else could we solve in polynomial time?

Reduction. Problem X **polynomial reduces to** problem Y if arbitrary instances of problem X can be solved using:
- Polynomial number of standard computational steps, plus
- Polynomial number of calls to oracle that solves problem Y.

Notation. \( X \leq_p Y. \)

Remarks.
- We pay for time to write down instances sent to black box \( \Rightarrow \) instances of Y must be of polynomial size.
- Note: Cook reducibility.

\( \text{computational model supplemented by special piece of hardware that solves instances of } Y \text{ in a single step} \)

\( \text{in contrast to Karp reductions} \)
**Polynomial-Time Reduction**

**Purpose.** Classify problems according to *relative* difficulty.

**Design algorithms.** If \( X \leq_p Y \) and \( Y \) can be solved in polynomial-time, then \( X \) can also be solved in polynomial time.

**Establish intractability.** If \( X \leq_p Y \) and \( X \) cannot be solved in polynomial-time, then \( Y \) cannot be solved in polynomial time.

**Establish equivalence.** If \( X \leq_p Y \) and \( Y \leq_p X \), we use notation \( X \equiv_p Y \).

\( \uparrow \)

up to cost of reduction
8.2 Reductions via "Gadgets"

Basic reduction strategies.

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction via "gadgets."
Satisfiability

Literal: A Boolean variable or its negation.

Clause: A disjunction of literals.

Conjunctive normal form: A propositional formula $\Phi$ that is the conjunction of clauses.

SAT: Given CNF formula $\Phi$, does it have a satisfying truth assignment?

3-SAT: SAT where each clause contains exactly 3 literals.

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

Yes: $x_1 = true$, $x_2 = true$, $x_3 = false$. 
3 Satisfiability Reduces to Independent Set

**Claim.** \(3\text{-SAT} \leq_p \text{INDEPENDENT-SET}.\)

**Pf.** Given an instance \(\Phi\) of 3-SAT, we construct an instance \((G, k)\) of INDEPENDENT-SET that has an independent set of size \(k\) iff \(\Phi\) is satisfiable.

**Construction.**
- \(G\) contains 3 vertices for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.
- Connect literal to each of its negations.

\[
\Phi = (\bar{x}_1 \lor x_2 \lor x_3) \land (x_1 \lor \bar{x}_2 \lor x_3) \land (\bar{x}_1 \lor x_2 \lor x_4)
\]
3 Satisfiability Reduces to Independent Set

Claim. $G$ contains independent set of size $k = |\Phi|$ iff $\Phi$ is satisfiable.

Pf. $\Rightarrow$ Let $S$ be independent set of size $k$.
- $S$ must contain exactly one vertex in each triangle.
- Set these literals to true.
- Truth assignment is consistent and all clauses are satisfied.

Pf $\Leftarrow$ Given satisfying assignment, select one true literal from each triangle. This is an independent set of size $k$.  

$\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)$
Review

Basic reduction strategies.
- Simple equivalence: INDEPENDENT-SET $\equiv_p$ VERTEX-COVER.
- Special case to general case: VERTEX-COVER $\leq_p$ SET-COVER.
- Encoding with gadgets: 3-SAT $\leq_p$ INDEPENDENT-SET.

Transitivity. If $X \leq_p Y$ and $Y \leq_p Z$, then $X \leq_p Z$.
Pf idea. Compose the two algorithms.

Ex: 3-SAT $\leq_p$ INDEPENDENT-SET $\leq_p$ VERTEX-COVER $\leq_p$ SET-COVER.
Self-Reducibility

Decision problem. Does there exist a vertex cover of size \( \leq k \)?

Search problem. Find vertex cover of minimum cardinality.

Self-reducibility. Search problem \( \leq_p \) decision version.

- Applies to all (NP-complete) problems in this chapter.
- Justifies our focus on decision problems.

Ex: to find min cardinality vertex cover.

- (Binary) search for cardinality \( k^* \) of min vertex cover.
- Find a vertex \( v \) such that \( G - \{v\} \) has a vertex cover of size \( \leq k^* - 1 \).
  - any vertex in any min vertex cover will have this property
- Include \( v \) in the vertex cover.
- Recursively find a min vertex cover in \( G - \{v\} \).

\[ \text{delete } v \text{ and all incident edges} \]
**Hamiltonian Cycle**

**HAM-CYCLE:** given an undirected graph $G = (V, E)$, does there exist a simple cycle $\Gamma$ that contains every node in $V$.

![Graph Diagram]

**NO:** bipartite graph with odd number of nodes.
**Directed Hamiltonian Cycle**

**DIR-HAM-CYCLE:** given a digraph $G = (V, E)$, does there exists a simple directed cycle $\Gamma$ that contains every node in $V$?

**Claim.** $\text{DIR-HAM-CYCLE} \leq_P \text{HAM-CYCLE}$. 

**Pf.** Given a directed graph $G = (V, E)$, construct an undirected graph $G'$ with $3n$ nodes.
Directed Hamiltonian Cycle

Claim. $G$ has a Hamiltonian cycle iff $G'$ does.

Pf. $\Rightarrow$

- Suppose $G$ has a directed Hamiltonian cycle $\Gamma$.
- Then $G'$ has an undirected Hamiltonian cycle (same order).

Pf. $\Leftarrow$

- Suppose $G'$ has an undirected Hamiltonian cycle $\Gamma'$.
- $\Gamma'$ must visit nodes in $G'$ using one of following two orders:
  - ..., B, G, R, B, G, R, B, G, R, B, ...
- Blue nodes in $\Gamma'$ make up directed Hamiltonian cycle $\Gamma$ in $G$, or reverse of one. $\blacksquare$
Claim. $\text{3-SAT} \leq_p \text{DIR-HAM-CYCLE}$. 

Pf. Given an instance $\Phi$ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamiltonian cycle iff $\Phi$ is satisfiable.

Construction. First, create graph that has $2^n$ Hamiltonian cycles which correspond in a natural way to $2^n$ possible truth assignments.
Construction. Given 3-SAT instance $\Phi$ with $n$ variables $x_i$ and $k$ clauses.

- Construct $G$ to have $2^n$ Hamiltonian cycles.
- Intuition: traverse path $i$ from left to right $\iff$ set variable $x_i = 1$. 
Construction. Given 3-SAT instance $\Phi$ with $n$ variables $x_i$ and $k$ clauses.

- For each clause: add a node and 6 edges.
3-SAT Reduces to Directed Hamiltonian Cycle

Claim. $\Phi$ is satisfiable iff $G$ has a Hamiltonian cycle.

Pf. $\Rightarrow$

- Suppose 3-SAT instance has satisfying assignment $x^*$.
- Then, define Hamiltonian cycle in $G$ as follows:
  - if $x^*_i = 1$, traverse row $i$ from left to right
  - if $x^*_i = 0$, traverse row $i$ from right to left
  - for each clause $C_j$, there will be at least one row $i$ in which we are going in "correct" direction to splice node $C_j$ into tour
Claim. $\Phi$ is satisfiable iff $G$ has a Hamiltonian cycle.

Pf. $\Leftarrow$

- Suppose $G$ has a Hamiltonian cycle $\Gamma$.
  - If $\Gamma$ enters clause node $C_j$, it must depart on mate edge.
    - thus, nodes immediately before and after $C_j$ are connected by an edge $e$ in $G$
    - removing $C_j$ from cycle, and replacing it with edge $e$ yields Hamiltonian cycle on $G - \{C_j\}$
  - Continuing in this way, we are left with Hamiltonian cycle $\Gamma'$ in $G - \{C_1, C_2, \ldots, C_k\}$.
  - Set $x^*_i = 1$ iff $\Gamma'$ traverses row $i$ left to right.
  - Since $\Gamma$ visits each clause node $C_j$, at least one of the paths is traversed in "correct" direction, and each clause is satisfied. $\blacksquare$
**Longest Path**

**SHORTEST-PATH.** Given a digraph $G = (V, E)$, does there exists a simple path of length at most $k$ edges?

**LONGEST-PATH.** Given a digraph $G = (V, E)$, does there exists a simple path of length at least $k$ edges?

**Claim.** $3$-SAT $\leq_p$ LONGEST-PATH.

**Pf 1.** Redo proof for DIR-HAM-CYCLE, ignoring back-edge from $t$ to $s$.
**Pf 2.** Show HAM-CYCLE $\leq_p$ LONGEST-PATH.
Traveling Salesperson Problem

**TSP.** Given a set of n cities and a pairwise distance function \(d(u, v)\), is there a tour of length \(\leq D\)?

All 13,509 cities in US with a population of at least 500
Reference:  http://www.tsp.gatech.edu
Traveling Salesperson Problem

**TSP.** Given a set of $n$ cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$?

Optimal TSP tour
Reference: http://www.tsp.gatech.edu
Traveling Salesperson Problem

TSP. Given a set of n cities and a pairwise distance function \( d(u, v) \), is there a tour of length \( \leq D \)?

11,849 holes to drill in a programmed logic array
Reference: http://www.tsp.gatech.edu
Traveling Salesperson Problem

**TSP.** Given a set of n cities and a pairwise distance function \( d(u, v) \), is there a tour of length \( \leq D \)?

Optimal TSP tour
Reference: http://www.tsp.gatech.edu
Traveling Salesperson Problem

**TSP.** Given a set of \( n \) cities and a pairwise distance function \( d(u, v) \), is there a tour of length \( \leq D \)?

**HAM-CYCLE:** given a graph \( G = (V, E) \), does there exists a simple cycle that contains every node in \( V \)?

**Claim.** \( \text{HAM-CYCLE} \leq_p \text{TSP}. \)

**Pf.**
- Given instance \( G = (V, E) \) of \( \text{HAM-CYCLE} \), create \( n \) cities with distance function
  \[
  d(u, v) = \begin{cases} 
  1 & \text{if } (u, v) \in E \\
  2 & \text{if } (u, v) \notin E 
  \end{cases}
  \]
- TSP instance has tour of length \( \leq n \) iff \( G \) is Hamiltonian.

**Remark.** TSP instance in reduction satisfies \( \Delta \)-inequality.