CMPSCI 311: Introduction to Algorithms

Lecture 22: Reductions and Intractability

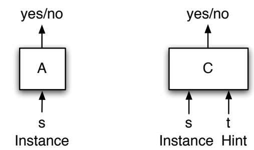
Dan Sheldon

University of Massachusetts

Last Compiled: April 24, 2017

Recap

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



Example

Problem (X)	INDEPENDENT-SET
Instance (s)	Graph G and number k
Algorithm (A)	Try all subsets and check (but not poly-time)
Hint (t)	Which nodes are in the answer?
Certifier (C)	Are those nodes independent and size k?

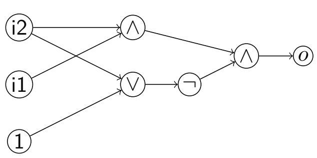
Proving New Problems NP-Complete

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 poly-time.
- Prove by reduction from CIRCUITSAT.

Example.



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 $\neg A \vee B$
  - B can additionally be a disjunction

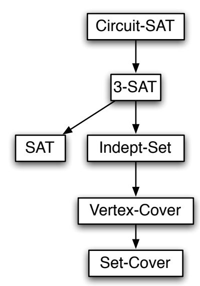
- Negation node: $x_v = \neg x_u$
  - $x_u \Rightarrow \neg x_v$
  - $\neg x_u \Rightarrow x_v$
- OR node: $x_v = x_u \vee x_w$
  - $x_u \Rightarrow x_v$
  - $x_w \Rightarrow x_v$
  - $x_v \Rightarrow x_u \vee x_w$
- AND node: $x_v = x_u \wedge x_w$
  - $x_v \Rightarrow x_u$
  - $x_v \Rightarrow x_w$
  - $\neg x_v \Rightarrow \neg x_u \vee \neg x_w$

- 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 satisfiable iff circuit is satisfiable.
- But it has clauses of size 1 and 2. Convert to 3-SAT formula by introducing two new variables and clauses that force them to be equal to zero.

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

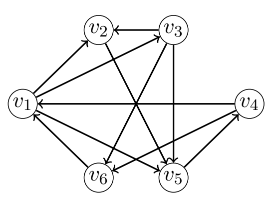
- Check $X \in$ NP.
- Choose known NP-complete problem Y.
- Prove $Y \leq_P X$.
- Usually suffices to do single transformation $s_Y \rightarrow s_X$ s.t.
  - s_X is YES instance iff s_Y is YES instance

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

- HAMCYCLE – 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?

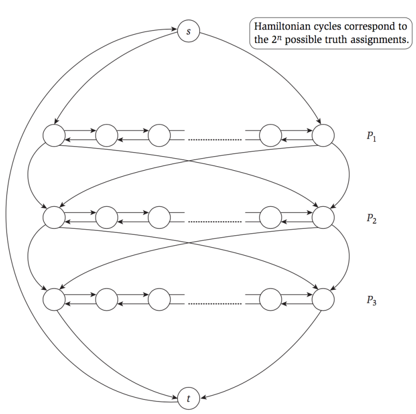
Claim. 3-SAT $\leq_P$ HAMCYCLE.

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



Reduction: Skeleton Construction

- 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_ℓ construct gadget to restrict possible truth assignments

- New node c_ℓ
- If $x_i \in C_\ell$
  - Add edges $(v_{i,3\ell}, c_\ell)$ and $(c_\ell, v_{i,3\ell+1})$
  - c_ℓ 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_ℓ 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 $L \to R$, else $R \to L$.
- Each C_ℓ is satisfied, so one path P_i is traversed in the correct direction to "splice" c_ℓ into our cycle
- The result is a Hamiltonian Cycle

Given Hamiltonian cycle, construct satisfying assignment:

- If cycle visits c_ℓ from row i, it will also leave to row i because of "buffer" nodes
- Therefore, ignoring clause nodes, cycle traverses each row completely from $L \to R$ or $R \to L$
- Set $x_i = 1$ if P_i traversed $L \to R$, 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

Reduction from HAM-CYCLE to TSP

Given HAMCYCLE instance $G = (V, E)$ make TSP instance

- One city per vertex
- $d(v_i, v_j) = 1$ if $(v_i, v_j) \in E$, else 2

Claim: there is a tour of distance $\leq n$ if and only if G has a Hamiltonian cycle

- A Hamiltonian cycle clearly gives a tour of length n
- A tour of length n must travel n hops of length 1, which corresponds to a Hamiltonian 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