CMPSCI 311: Introduction to Algorithms

Lecture 20: Reductions and Intractability

Akshay Krishnamurthy

University of Massachusetts

Last Compiled: April 23, 2018

Announcements

- Quiz due tonight
- HW 6 due 5/1 (Tuesday night!), and extra credit
- Midterms back on wednesday (Solutions up tonight)
- Last discussion on friday
- Final Exam: Friday 5/4, 3:30-5:30pm, Marcus Hall 131.

Recap

- Problem X is a set of strings s, the YES instances.
- Algorithm A solves X if $A(s) = \textsc{true}$ 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 $|t| \le p(|s|)$ and $B(s,t) = \textsc{True}$.
- $\mathcal{P}$ – class of problems with polytime algorithm.
- $\mathcal{NP}$ – class of problems with polytime certifier.
- X is **NP-Complete** iff $Y \le_P X$ for all $Y \in \mathcal{NP}$.

Example

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

Plan for today

- Review 3-SAT $\le_P$ CircuitSat
- HamCycle
- TSP

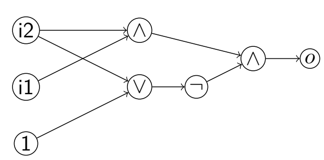
Back to 3-SAT

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

Theorem. 3-SAT is NP-Complete.

- Clearly in $\mathcal{NP}$.
- Prove by reduction from CircuitSAT.

Example.



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 $(x_v \vee x_u), (\neg x_v \vee \neg x_u)$.
  - If v is $\vee$ with u, w incoming then $(x_v \vee \neg x_u), (x_v \vee \neg x_w), (\neg x_v \vee x_u \vee x_w)$.
  - If v is $\wedge$ then $(\neg x_v \vee x_u), (\neg x_v \vee x_w), (x_v \vee \neg x_u \vee \neg x_w)$.
- Input bits get set with (x_v) if fixed to one and $(\neg x_v)$ otherwise.
- Clause (x_o) for output bit.

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. $\textsc{IndependentSet}$, $\textsc{VertexCover}$, $\textsc{SetCover}$, 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.

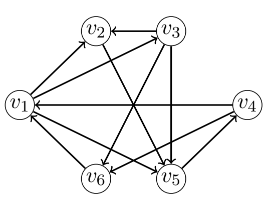
Two new problems.

- 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?

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

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

$\textsc{HamCycle}$ Example



$\textsc{HamCycle}$

Theorem. $\textsc{HamCycle}$ is NP-Complete.

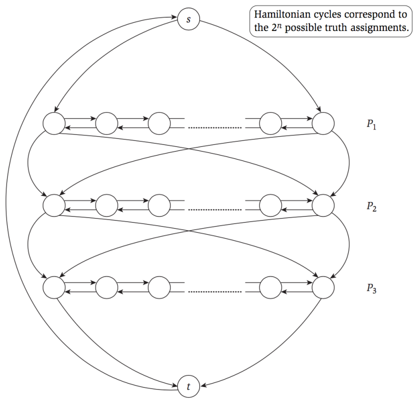
- It is in $\mathcal{NP}$.
- Need to reduce from some NP-Complete problem. Which one?

Claim. $\text{3-SAT} \leq_P \textsc{HamCycle}$.

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 $4m + 2$ 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 $C_\ell = x_i \vee \neg x_j \vee x_k$ new node c_ℓ in graph.

- Edges $(v_{i,4\ell}, c_\ell)$ and $(c_\ell, v_{i,4\ell+1})$.
- Edges $(v_{j,4\ell+1}, c_\ell)$ and $(c_\ell, v_{j,4\ell})$.
- Edges $(v_{k,4\ell}, c_\ell)$ and $(c_\ell, v_{k,4\ell+1})$.

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

Example

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

Proof

If ϕ is satisfying assignment

- If $\phi(x_i) = 1$ 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?

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

Theorem. TSP is NP-Complete

- Clearly in $\mathcal{NP}$.
- Reduction from $\textsc{HamCycle}$.

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. (assymetric).
- 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 $f : V \to \{1, \ldots, k\}$ 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.

Graph Coloring

Claim. 2-coloring $\in \mathcal{P}$.

Proof.

- 2-coloring equivalent to bipartite testing.

Theorem. 3-coloring is NP-Complete.

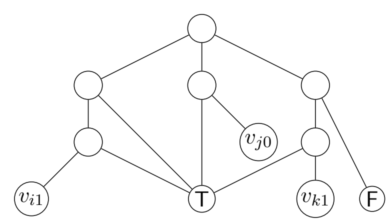
Reduction

Reduce from 3-SAT.

- Skeleton – Idea: 1 color for True, 1 for False
  - 3 extra nodes in a clique T, F, B.
  - For each variable x_i, two nodes v_{i0}, v_{i1}.
  - Edges $(v_{i0}, B), (v_{i1}, B), (v_{i0}, v_{i1})$.
- Either v_{i0} or v_{i1} gets the T color.

Reduction

For clause $x_i \vee \neg x_j \vee x_k$



Proof

- Graph is polynomial in $n + m$.
- If satisfying assignment
  - Color B, T, F then v_{i1} as T if $\phi(x_i) = 1$.
  - Since clauses satisfied, can color each gadget.
- If graph 3-colorable
  - One of v_{i0}, v_{i1} must get T color.
  - Clause gadget colorable iff clause satisfied.

Question. What about k-coloring?