COMPSCI 311: Introduction to Algorithms

Lecture 22: Reductions and NP-Complete Problems

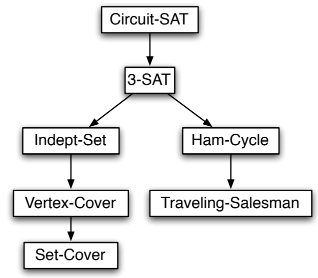
Marius Minea

University of Massachusetts Amherst

slides credit: Dan Sheldon (adapted)

NP-Complete Problems So Far

Theorem: INDEPENDENTSET, VERTEXCOVER, SETCOVER, SAT, 3-SAT, HAM-CYCLE, HAM-PATH, TSP are all NP-Complete.



Arrows show reductions discussed in class.
We could construct a polynomial reduction between any pair.

Intractability: quiz 1

Which of the following graph problems are known to be in NP?

A. Is the length of the longest simple path $\leq k$?

B. Is the length of the longest simple path $\geq k$?

C. Is the length of the longest simple path $= k$?

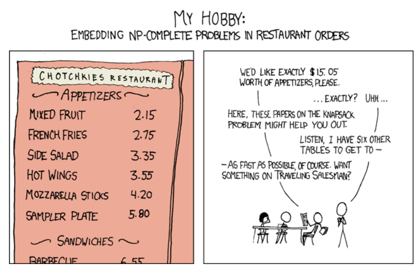
D. Find the length of the longest simple path.

E. All of the above.

slide credit: Kevin Wayne / Pearson

Numerical problems

Subset Sum decision problem: given n items with weights $w_1, \dots, w_n$, is there a subset of items whose weight is exactly W?



Dynamic programming: $O(nW)$ *pseudo-polynomial* time algorithm (not polynomial in input length $n \log W$)

Subset Sum

Theorem. Subset sum is NP-complete.

Reduction from 3-SAT. (n variables, m clauses, base 10).

- All weights have $n + m$ digits
- Digits 1 to n: For variable x_i, create two items t_i, f_i
  - Both have ith digit equal to 1
  - All other items have zero in this digit
  - ith digit of $W = 1 \Rightarrow$ select exactly one of t_i, f_i
- The $n + j$th digit corresponds to clause C_j
  - If $x_i \in C_j$, set $n + j$th digit of $t_i = 1$
  - If $\neg x_i \in C_j$, set $n + j$th digit of $f_i = 1$
  - Everything else 0.

Subset Sum (cont.)

- Set $n + j$th digit of $W = 3$
  - Consider a subset of items corresponding to a truth assignment (exactly one of t_i, f_i)
  - If C_j is not satisfied, then total in position $n + j$ is 0, otherwise it is 1, 2, or 3
  - Create two "dummy" items y_j, z_j with 1 in position $n + j$
  - Can select dummies to yield total of 3 in position $n + j$ iff C_j is satisfied

Subset Sum Example

Example.

$$(x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)$$

Item	1	2	3	4	5	6
t_1	1	0	0	1	0	0
f_1	1	0	0	0	1	1
t_2	0	1	0	0	1	0
f_2	0	1	0	1	0	1
t_3	0	0	1	1	0	1
f_3	0	0	1	0	1	0
W	1	1	1	3	3	3

Item	1	2	3	4	5	6
y_1	0	0	0	1	0	0
z_1	0	0	0	1	0	0
y_2	0	0	0	0	1	0
z_2	0	0	0	0	1	0
y_3	0	0	0	0	0	1
z_3	0	0	0	0	0	1

Subset Sum Proof

- All numbers (including W) are polynomially long.
- If Φ satisfiable,
  - Select t_i if $x_i = 1$ in satisfying assignment else select f_i.
  - Take y_j, z_j as needed.
- If subset exists with sum W
  - Either t_i or f_i is chosen. Assign x_i accordingly.
  - For each clause, at least one term must be selected, otherwise clause digit is < 3.

Warning

Theorem. SubsetSum is NP-Complete.

But Subset Sum can be tricky!

- If reducing SubsetSum $\leq_P X$, reduction needs to be polynomial in $\log(W)$ (number of digits).

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.

Claim. 2-coloring $\in$ P (equivalent to bipartite testing)

Theorem. 3-coloring is NP-Complete.

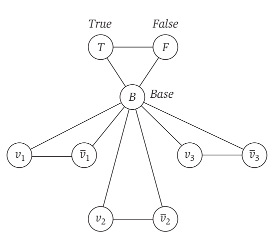
Reduction

- Reduce from 3-SAT.

Skeleton: 1 color for True, 1 for False

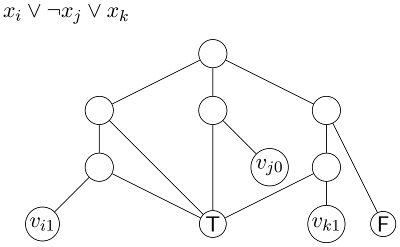
3 extra nodes in a clique T, F, B.
For each variable x_i, two nodes v_{i0}, v_{i1}.
Edges $(v_{i0}, B), (v_{i1}, B), (v_{i0}, v_{i1})$.

Either v_{i0} or v_{i1} gets the T color.



Reduction

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



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?

Clicker Question 2

Which of the following is true?

A: If we can reduce 3-coloring to k-coloring, then k-coloring is NP-complete

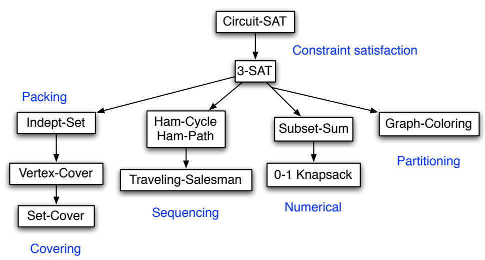
B: k-coloring is NP-complete since any 3-coloring is also a k-coloring for $k \geq 3$

C: k-coloring is not NP-complete since 3-coloring is the hardest case, for $k > 3$ the coloring is easier

D: k-coloring is not NP-complete because the 4-color theorem has been proved

NP-Completeness Recap

Types of hard problems:



. . . any many others. See book or other sources for more examples. You can use *any known NP-complete* problem to prove a new problem is NP-complete.