Intro to Theory of Computation

LECTURE 24

CS 332

Last time

- Polynomial-time reductions
- NP-completeness

Today

- Cook-Levin Theorem
- Examples of NP-complete languages

Sofya Raskhodnikova

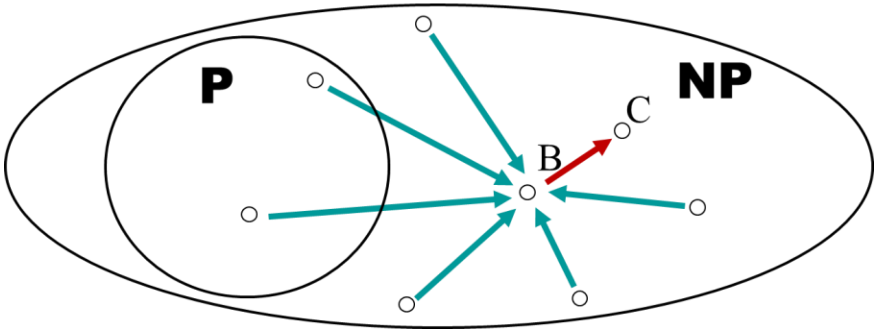
CS 332

Implication of poly-time reductions

Theorem. If

- B is **NP**-complete,
- C$\in$ **NP** and
- B$\leq_p$C

then C is **NP**-complete.



Theorem. If B is **NP**-complete and B$\in$ **P** then

$$\mathbf{P} = \mathbf{NP}.$$

(So, if B is **NP**-complete and **P** $\neq$ **NP** then there is no poly-time algorithm for B.)

CS 332

The class NP

NP is the class of languages that have polynomial-time verifiers.

CS 332

Question

Recall: The running time of a verifier V($\langle w, c \rangle$) is measured only in terms of length of w.

If we allowed a verifier to run in time polynomial in the length of $\langle w, c \rangle$, the class NP would

A. **be smaller**

B. **be the same**

C. **contain more (decidable only) languages**

D. **contain more (even undecidable) languages**

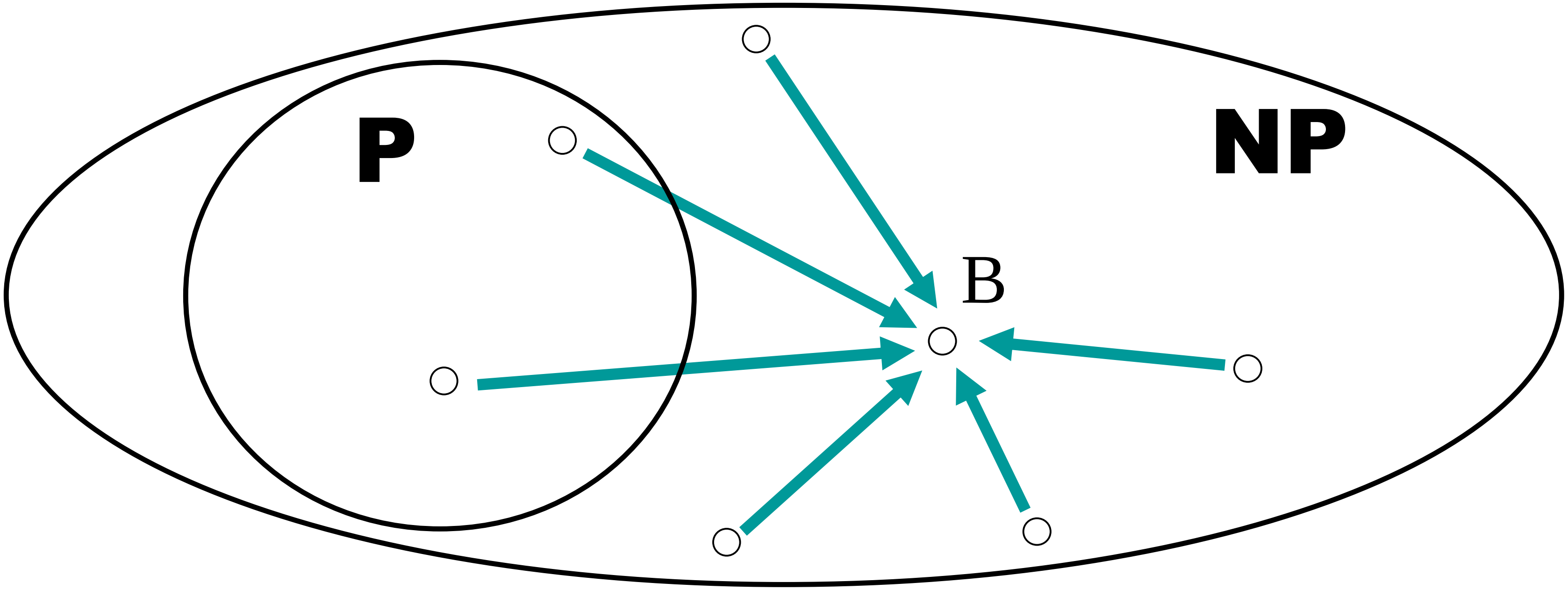
E. **none of the above**

CS 332

Hardest problems in NP

A language B is **NP-complete** if

1. B∈ **NP**
2. B is **NP-hard**,i.e.,
   every language in NP is poly-time reducible to B.



An NP-complete problem

BA_{NTM} = {⟨M,x,$\underline{t}$⟩ | M is an NTM that accepts x in at most t steps}

Technical detail: $\underline{n}$ denotes 1^n.

Theorem. BA_{NTM} is NP-Complete.

1. $BA_{NTM} \in NP$:

   The list of guesses M makes to accept x in t steps is the certificate that $\langle M,x,\underline{t}\rangle \in BA_{NTM}$.

2. For all $A \in NP$, $A \leq_P BA_{NTM}$.

   $A \in NP$ iff there is an NTM N for A that runs in time $O(n^k)$.
   Let $f_A(w) = \langle N,w,\underline{c\,|w|^k}\rangle$.
   $\langle N,w,\underline{c\,|w|^k}\rangle \in BA_{NTM} \iff$ N accepts w $\iff w \in A$.

The "First" NP-Complete Problem

CIRCUIT-SAT is similar to 3SAT, but it is about circuits, not formulas.

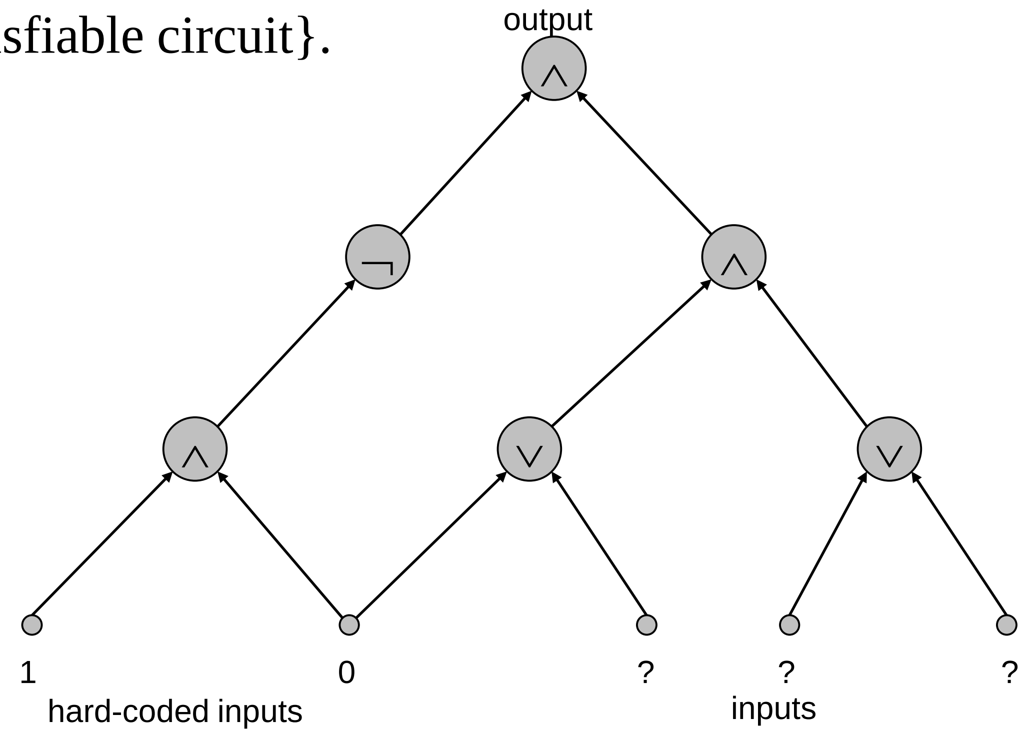
Theorem. CIRCUIT-SAT is NP-complete. [Cook `71, Levin `73]

A **circuit** is built from AND, OR and NOT gates.

A circuit is **satisfiable** if one can set the circuit inputs, so that the output is 1.

CIRCUIT-SAT = $\{\langle C \rangle | \ C$ is a satisfiable circuit$\}$.

yes: 1 0 1



CS 332

Canonical NP-Complete Problem

Theorem. 3SAT is NP-complete. (Book)

Establishing NP-completeness

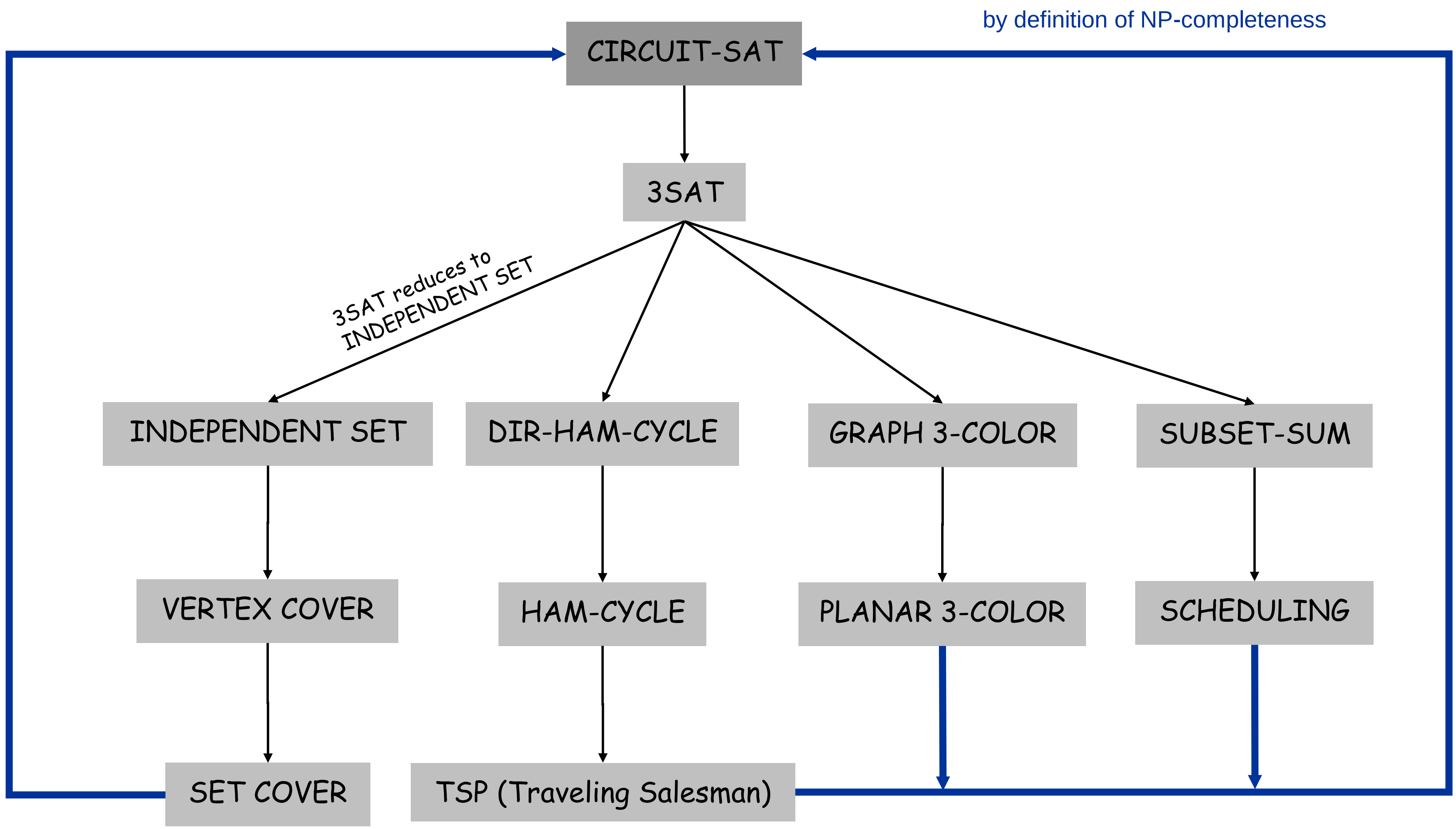
- Once we establish first "natural" NP-complete problems, others fall like dominoes.

- Recipe to establish NP-completeness of problem Y.
  - Step 1. Show that Y is in NP.
  - Step 2. Choose an NP-complete problem X (e.g., 3SAT) and prove that $X \leq_p Y$.

CS 332

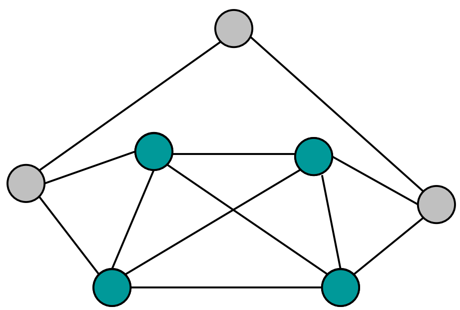
NP-Completeness

All problems below are NP-complete and hence poly-time reduce to one another!



CLIQUE

- A clique in a graph is a set of nodes, where every two nodes are connected by an edge.



- CLIQUE = $\{\langle G, k\rangle \mid G$ is an undirected graph that contains a clique with k nodes$\}$

Prove: CLIQUE is NP-complete

- CLIQUE = $\{\langle G, k\rangle \mid G$ is an undirected graph that contains a clique with k nodes$\}$

1. Show that CLIQUE is in NP.

   Certificate: clique of size k

2. Show that 3SAT $\leq_p$ CLIQUE.

   "On input $\langle \phi \rangle$, where ϕ is a 3cnf formula,
   Output a graph G and a number k, where… "

(on the board)

SUBSET-SUM

- SSUM = $\{\langle S, t\rangle \mid S = \{x_1, \dots, x_r\}$,
  and for some $\{y_1, \dots, y_{r'}\} \subseteq \{x_1, \dots, x_r\}$,
  we have $y_1 + \cdots + y_{r'} = t\}$

- Examples: $\langle\{5, 7, 23\}, 28\rangle \in SSUM$
  $\langle\{5, 7, 25\}, 28\rangle \notin SSUM$

CS 332 Prove: *SSUM* is NP-complete

- SSUM = $\{\langle S, t\rangle \mid S = \{x_1, \ldots, x_r\},$ and for some $\{y_1, \ldots, y_{r'}\} \subseteq \{x_1, \ldots, x_r\},$ we have $y_1 + \cdots + y_{r'} = t\}$

1. Show that $SSUM$ is in NP.

   Certificate: subset $y_1, \ldots, y_{r'}$ of S that sums up to t.

2. Show that 3SAT $\leq_p SSUM$.

“On input $\langle \phi \rangle$, where ϕ is a 3cnf formula,

with variables $x_1, \ldots, x_\ell$ and clauses $c_1, \ldots, c_k$

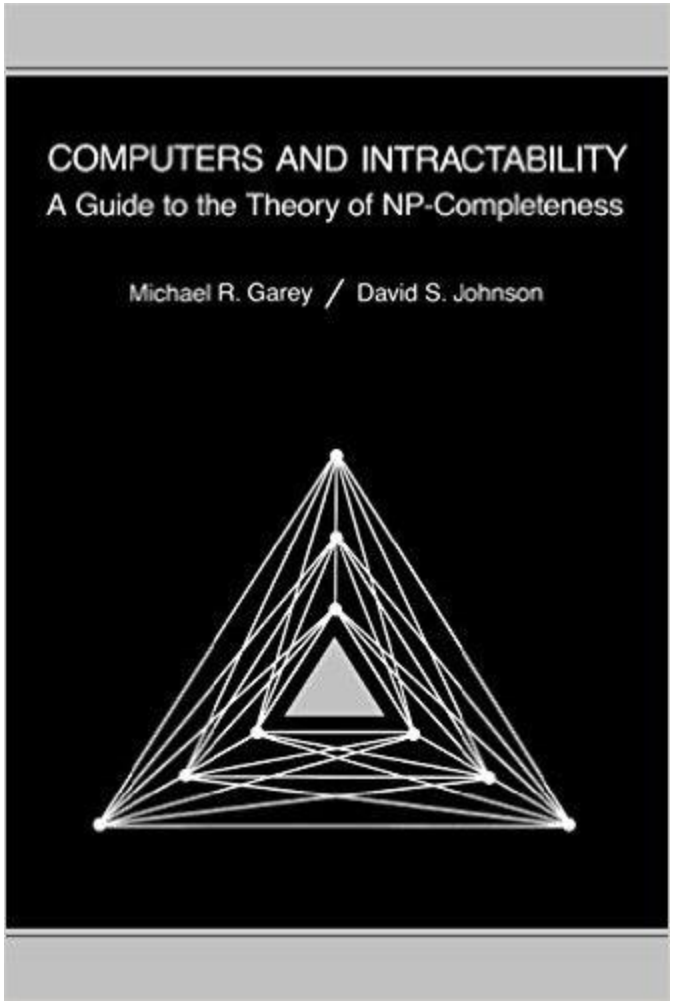
- Output a set S of numbers and a target number $t \ldots$ ”

(on the board)

Some NP-Complete Problems

- Six basic genres of NP-complete problems and paradigmatic examples.
  - Packing problems: SET-PACKING, INDEPENDENT SET.
  - Covering problems: SET-COVER, VERTEX-COVER.
  - Constraint satisfaction problems: SAT, 3-SAT.
  - Sequencing problems: HAMILTONIAN-CYCLE, TSP.
  - Partitioning problems: 3D-MATCHIN,G 3-COLOR.
  - Numerical problems: SUBSET-SUM, KNAPSACK.
- Most NP problems are either known to be in P or NP-complete.
- Notable exceptions. Factoring, graph isomorphism, Nash equilibrium.

An encyclopedia of NP-complete problems



CS 332

From Garey and Jonhson

``We need an efficient algorithm that constructs a design of iThingy that meets the maximum # of requirements at lowest cost.”

``I thought about it for weeks, but I can’t come up with an efficient algorithm.
I guess I’m just too dumb.”

“I can’t find an efficient algorithm, but neither can all these famous people.”

Question

3SAT is an example of a problem that cannot be solved by an algorithm.

A. True

B. False

C. It is an open question

D. None of the above

Question

3SAT is an example of a problem that cannot be solved by a polynomial time algorithm.

A. True

B. False

C. It is an open question

D. None of the above