CS 381 – FALL 2019

Week 14.1, Monday, Nov 18

Homework 7 Due: November 26th at 11:59PM on Gradescope

Q1b Typo: if there is a directed from v_ℓ to $v_{\bar{\ell}}$ AND from $v_{\bar{\ell}}$ to v_ℓ then the 2-SAT instance is not satisfiable.

8.4 NP-Completeness

NP-Complete

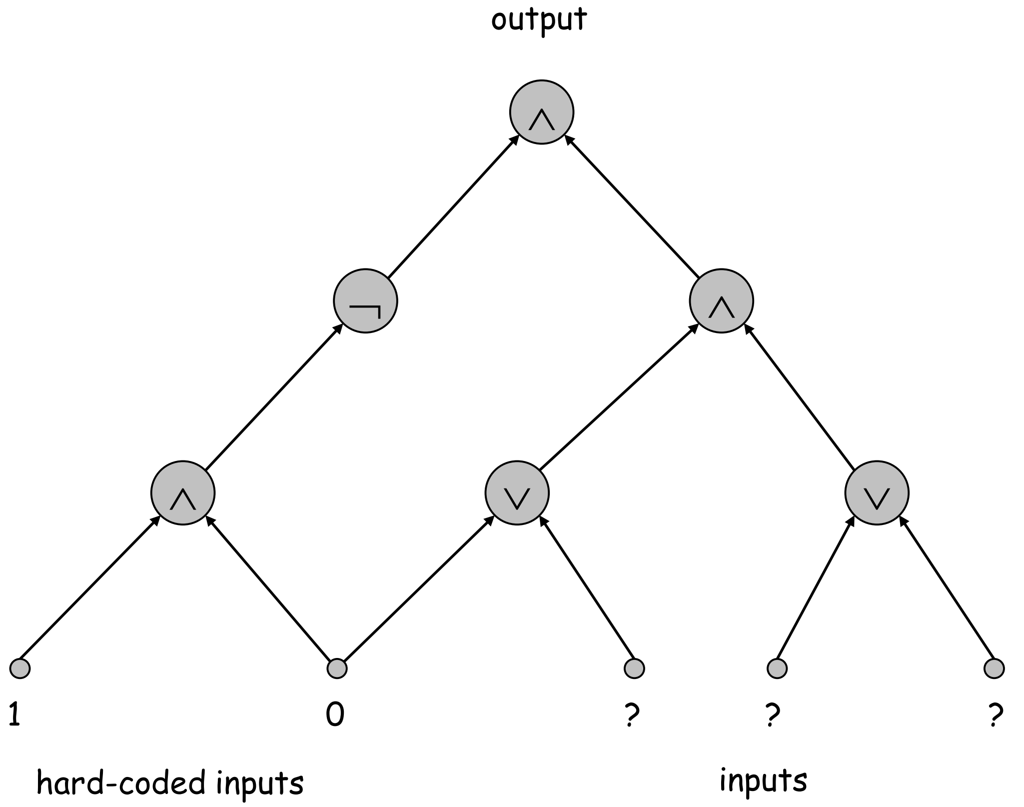
NP-complete. A problem Y in NP with the property that for every problem X in NP, $X \leq_p Y$.

Theorem. Suppose Y is an NP-complete problem. Then Y is solvable in poly-time iff P = NP.

Fundamental question. Do there exist "natural" NP-complete problems?

Circuit Satisfiability

CIRCUIT-SAT. Given a combinational circuit built out of AND, OR, and NOT gates, is there a way to set the circuit inputs so that the output is 1?



yes: 1 0 1

The "First" NP-Complete Problem

Theorem. CIRCUIT-SAT is NP-complete. [Cook 1971, Levin 1973]

Pf. (sketch)

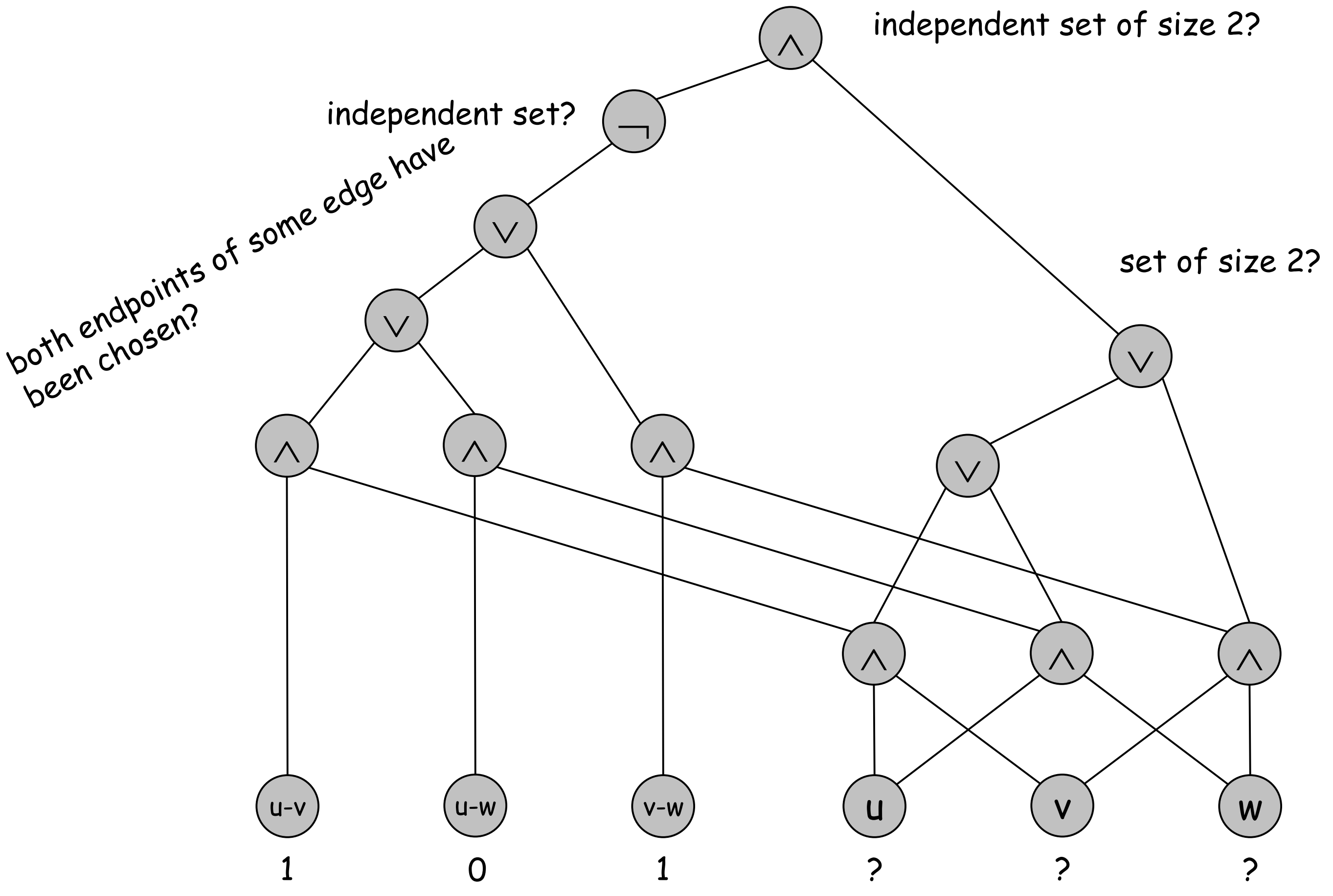
- Any algorithm that takes a fixed number of bits n as input and produces a yes/no answer can be represented by such a circuit. Moreover, if algorithm takes poly-time, then circuit is of poly-size.

  sketchy part of proof; fixing the number of bits is important,
  and reflects basic distinction between algorithms and circuits

- Consider some problem X in NP. It has a poly-time certifier C(s, t). To determine whether s is in X, need to know if there exists a certificate t of length p(|s|) such that C(s, t) = yes.
- View C(s, t) as an algorithm on |s| + p(|s|) bits (input s, certificate t) and convert it into a poly-size circuit K.
  - first |s| bits are hard-coded with s
  - remaining p(|s|) bits represent bits of t
- Circuit K is satisfiable iff there exists a length p(|s|) bit input string t s.t. C(s, t) = yes.

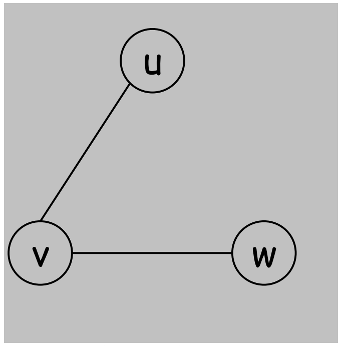
Example

Ex. Construction below creates a circuit K whose inputs can be set so that K outputs true iff graph G has an independent set of size 2.



$\binom{n}{2}$ hard-coded inputs (graph description)

n inputs (nodes in independent set)



G = (V, E), n = 3

Establishing NP-Completeness

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

Recipe to establish NP-completeness of problem Y.

- Step 1. Show that Y is in NP.
- Step 2. Choose a known NP-complete problem X.
- Step 3. Prove that $X \le_p Y$.

Justification. If X is an NP-complete problem, and Y is a problem in NP with the property that $X \le_P Y$ then Y is NP-complete.

Pf. Let W be any problem in NP. Then $W \underset{\substack{\uparrow \\ \text{by definition of} \\ \text{NP-complete}}}{\le_P} X \underset{\substack{\uparrow \\ \text{by assumption}}}{\le_P} Y$.

- By transitivity, $W \le_P Y$.
- Hence Y is NP-complete. ▪

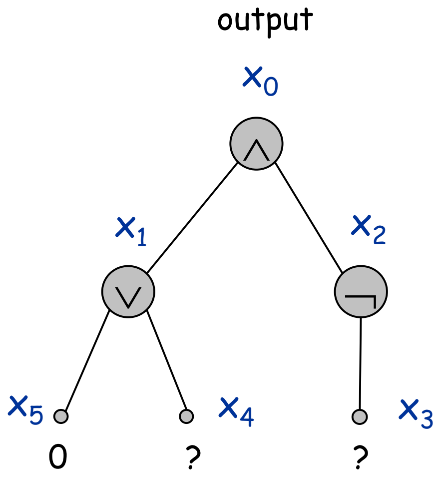
3-SAT is NP-Complete

Theorem. 3-SAT is NP-complete.

Pf. Suffices to show that CIRCUIT-SAT $\leq_P$ 3-SAT since we previously showed 3-SAT is in NP.

- Let K be any circuit we build a 3-SAT formula $\Phi_K = f(K)$.
- Create a 3-SAT variable x_i for each circuit element i.
- Make circuit compute correct values at each node:
  - $x_2 = \neg x_3$ $\Rightarrow$ add 2 clauses: $x_2 \vee x_3$, $\overline{x_2} \vee \overline{x_3}$
  - $x_1 = x_4 \vee x_5$ $\Rightarrow$ add 3 clauses: $x_1 \vee \overline{x_4}$, $x_1 \vee \overline{x_5}$, $\overline{x_1} \vee x_4 \vee x_5$
  - $x_0 = x_1 \wedge x_2$ $\Rightarrow$ add 3 clauses: $\overline{x_0} \vee x_1$, $\overline{x_0} \vee x_2$, $x_0 \vee \overline{x_1} \vee \overline{x_2}$

- Hard-coded input values and output value.
  - $x_5 = 0$ $\Rightarrow$ add 1 clause: $\overline{x_5}$
  - $x_0 = 1$ $\Rightarrow$ add 1 clause: x_0
- Final step: turn clauses of length < 3 into clauses of length exactly 3.
  - Add new variable z and new clause $(\bar{z} \vee \bar{z} \vee \bar{z})$
  - Pad short clauses (len < 3) with z e.g. $\overline{x_5} \vee z \vee z$.



If this was a homework set…

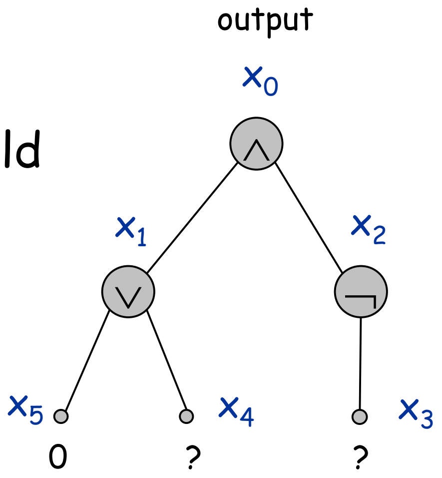
Theorem. 3-SAT is NP-complete.

Pf. Suffices to show that CIRCUIT-SAT $\le_P$ 3-SAT since we previously showed 3-SAT is in NP.

- Let K be any circuit we built a 3-SAT formula $\Phi_K = f(K)$.

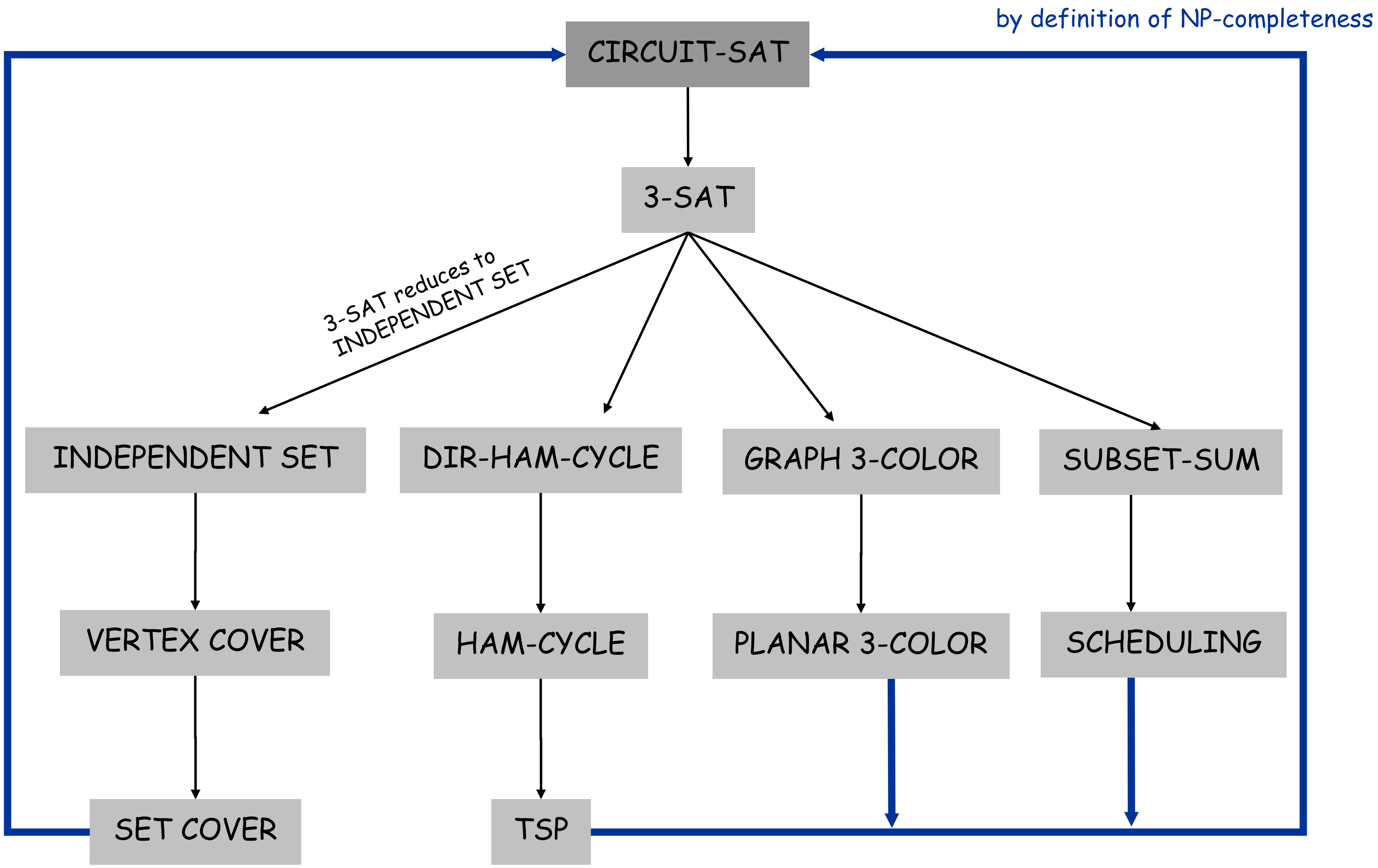
1. Explain why $\Phi_K = f(K)$ can be constructed in polynomial time.
2. Show that if K has a satisfying assignment then $\Phi_K = f(K)$ has a satisfying assignment.
3. Show that if K does not have a satisfying assignment then $\Phi_K = f(K)$ does not have a satisfying assignment

(Easier to show contrapositive of step 3)
Assume $\Phi_K = f(K)$ has a satisfying assignment and then build satisfying assignment for circuit K.



NP-Completeness

Observation. All problems below are NP-complete and polynomial reduce to one another!



iClicker

Jane is excited! She thinks she has proved that P=NP. In particular, she claims to have proved that 2-SAT is NP-Complete by showing that 2SAT$\leq_p$3SAT. She has a reduction f which maps 2-SAT instances (Φ_{2SAT}) to 3-SAT instances (Φ_{3SAT}) in polynomial time. She proved that if Φ_{2SAT} is satisfiable then $\Phi_{3SAT} = \mathrm{f}(\Phi_{2SAT})$ is satisfiable. What mistakes (if any) did Jane make?

A. Proving that 2-SAT is NP-Complete would not imply P=NP.
B. Jane still needs to show that if Φ_{3SAT} is satisfiable then Φ_{2SAT} is satisfiable to conclude that 2SAT$\leq_p$3SAT .
C. The reduction is in the wrong direction to conclude that P=NP.
D. Claims B and C are both true.
E. Jane's logic is sound! She can claim the $1,000,000 prize!

iClicker

Jane is excited! She thinks she has proved that P=NP. In particular, she claims to have proved that 2-SAT is NP-Complete by showing that 2SAT$\leq_p$3SAT. She has a reduction f which maps 2-SAT instances (Φ_{2SAT}) to 3-SAT instances (Φ_{3SAT}) in polynomial time. She proved that if Φ_{2SAT} is satisfiable then $\Phi_{3SAT} = \mathrm{f}(\Phi_{2SAT})$ is satisfiable. What mistakes (if any) did Jane make?

A. Proving that 2-SAT is NP-Complete would not imply P=NP.
  - 2SAT is in P (see homework 7). This would imply P=NP.

B. Jane still needs to show that if Φ_{3SAT} is satisfiable then Φ_{2SAT} is satisfiable to conclude that 2SAT$\leq_p$3SAT.
  - Otherwise $\mathrm{f}(\Phi_{2SAT})$ could ignore the input Φ_{2SAT} and *always* output a satisfiable 3SAT formula

C. The reduction is in the wrong direction to conclude that P=NP.
  - Jane would need to prove that 3SAT$\leq_p$2SAT

D. Claims B and C are both true.

E. Jane's logic is sound! She can claim the $1,000,000 prize!

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-MATCHING, 3-COLOR.
- Numerical problems: SUBSET-SUM, KNAPSACK.

Practice. Most NP problems are either known to be in P or NP-complete.

Notable exceptions. Factoring, graph isomorphism, Nash equilibrium.