Learning Goals

- ✓ Define NP-complete and NP-Hard Problems and describe their importance
- ✓ Describe parts of NP-Complete Proof
- Practice proving a problem is NP complete (Hamiltonian Path)

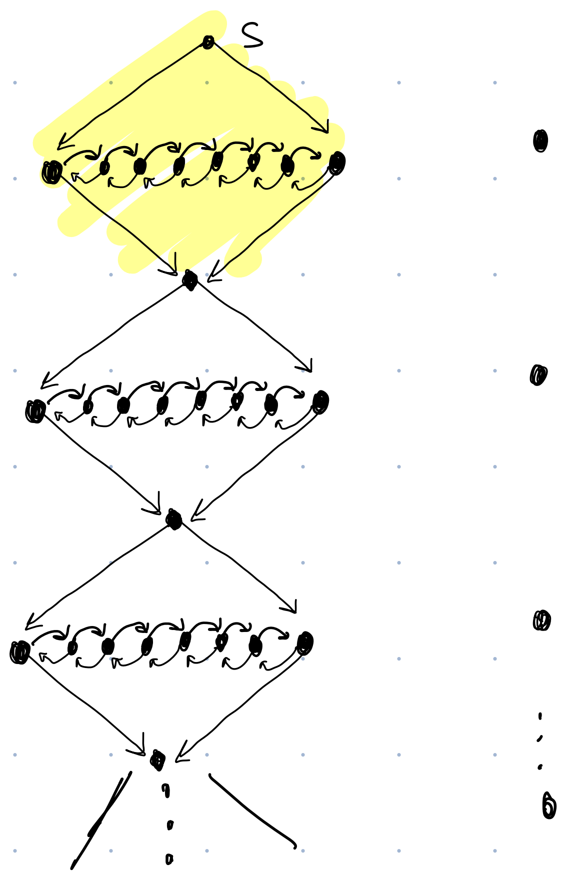
Exit Tickets

SAT Competition

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

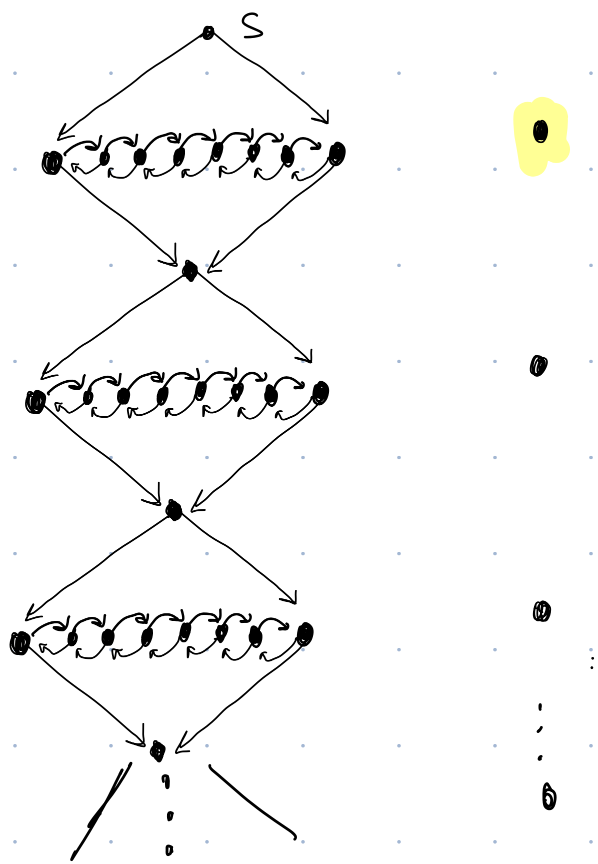
3 variables: x_1, x_2, x_3

How Many Gadgets? $\boxed{\text{A: } 3}$ B: 4



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

How many non-gadget vertices? A: 3 $\boxed{\text{B: 4}}$



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

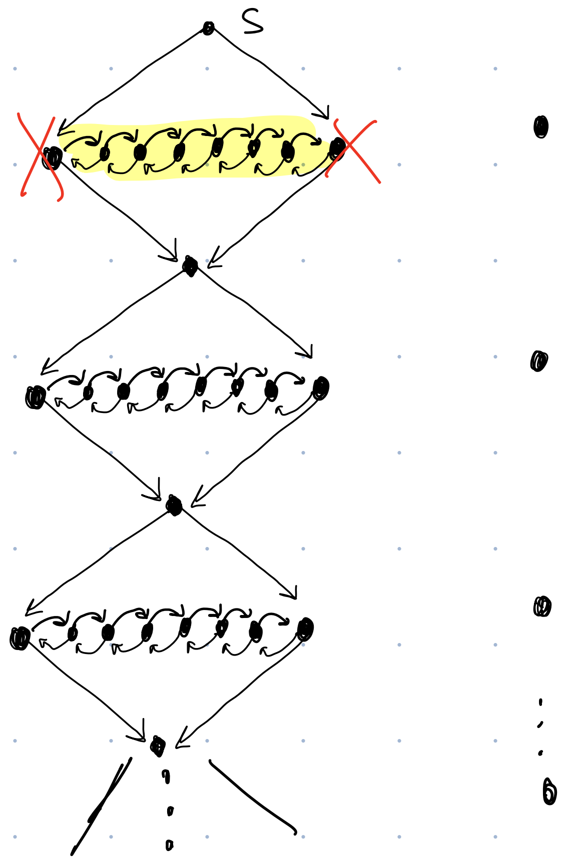
How Many horizontal vertices?

A: 3 B: 4

C: 6 $\boxed{\text{D: 8}}$

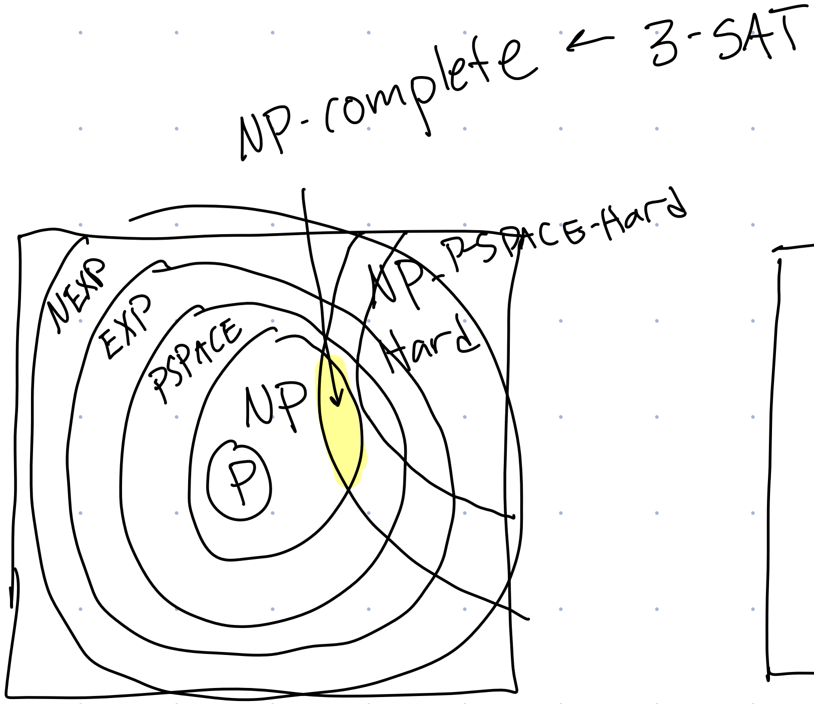
TRUE

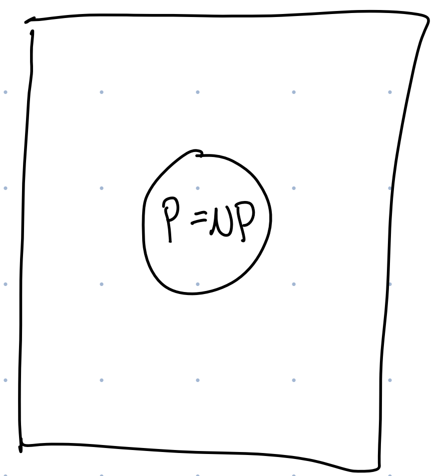
False



Review

NP NP-Hard NP-Complete





Types of Problems

Easy
(Polynomial time)

- Search
- Sort
- Multiplication
- Closest Points
- Greedy Scheduling
- MWIS on a line
- Matrix Mult.

Puzzles / NP

Crossword

Sudoku

Delivery rt ≤ 100 miles

Protein Folding

Factor larger numbers

Primality Testing

Quantum

Reg.

Question: How do we identify the hardest problems in NP?

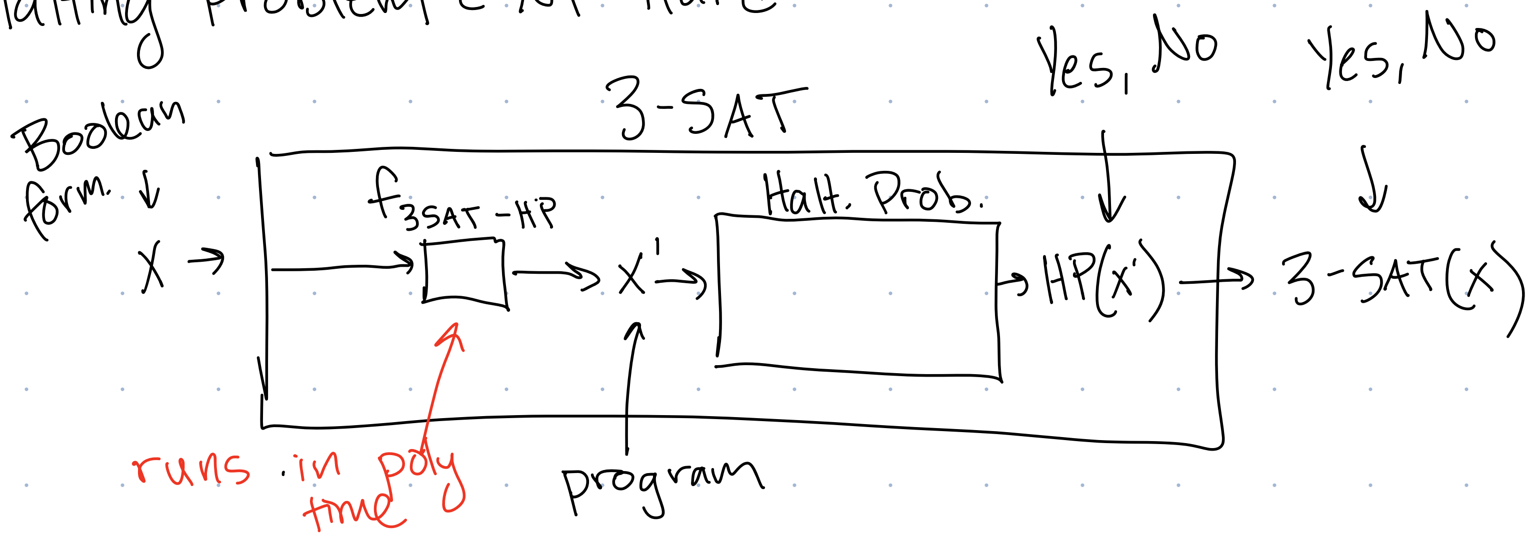
→ Empirical: Keep trying to find alg... but can't... HARD

→ Analytical: Prove a problem is hard. Possible!

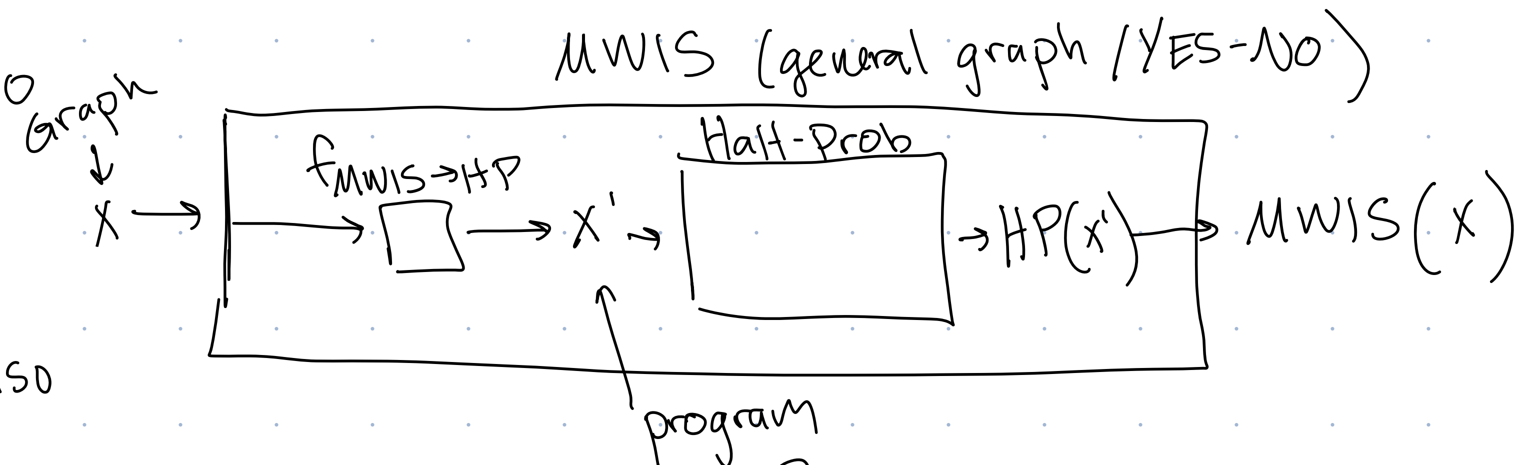
NP-Hard

def: A problem $Q \in$ NP-Hard if for every problem $R \in NP$, $R \leq_p Q$.

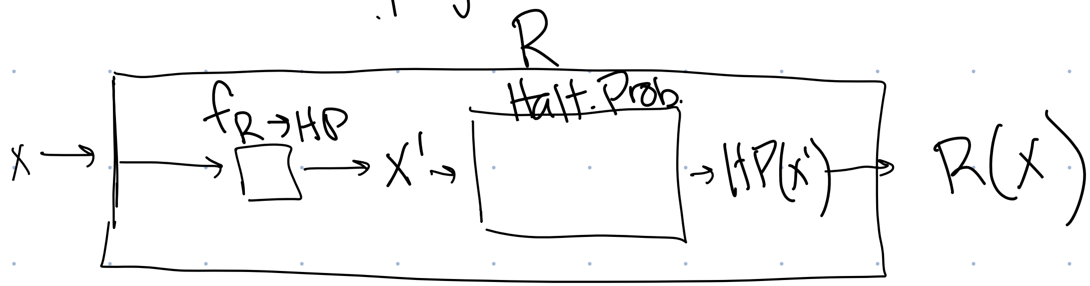
Ex: Halting Problem $\in$ NP-Hard



Also

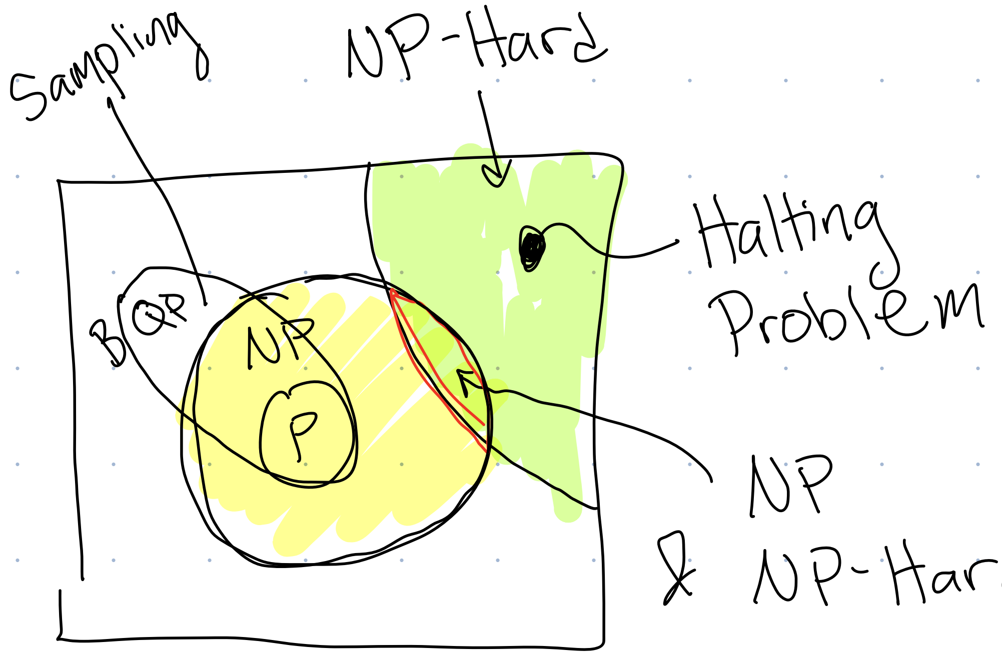


Also



NP-Hard problem are harder / require more resources
→ than NP problems, b/c if could solve, then would have power to solve any NP problem.

<u>BUT</u>



NP & NP-Hard } def: NP-Complete

↑

Hardest problems in NP:
(Traveling Salesperson, MWIS, Negative cycle avoiding shortest path)

<u>def</u>: $Q \in$ NP-Complete if $Q \in NP$ and $Q \in$ NP-Hard

$\boxed{\text{Fact 1: 3SAT} \in \text{NP-Hard}}$ (see 301)

$\boxed{\text{Lemma 1: If } Q \in \text{NP-Hard and } Q \leq_p R \text{ then } R \in \text{NP-Hard}}$

(pset 10)

Theorem: Hamiltonian Path is NP-Complete

Pf
- Ham-Path $\in$ NP [insert proof from NP class]
- Ham-Path $\in$ NP-Hard
  - 3SAT $\leq_p$ Ham-Path

$\downarrow$ by Fact 1 + Lemma 1

Ham-Path $\in$ NP-Hard

Formal Definition of Polytime Reduction

<u>def</u>: $R \leq_p Q$ ("R is polytime reducible to Q") if

- ✓ $\exists\ f_{R\to Q} : \{0,1\}^* \to \{0,1\}^*$ s.t.
- $\exists$ constant $c_{R\to Q}$ s.t. runtime of $f_{R\to Q}$ on input x is $O(|x|^{c_{R\to Q}})$ (Polytime)
- $\forall\ x \in \{0,1\}^*$, $R(x) = \text{Yes}$ iff $Q(f_{R\to Q}(x)) = \text{Yes}$

(Correctly convert input)

Lemma: $3SAT \leq_p$ Ham-Path

Strategy: (1.) Describe $f_{3SAT \to \text{Ham-Path}}$

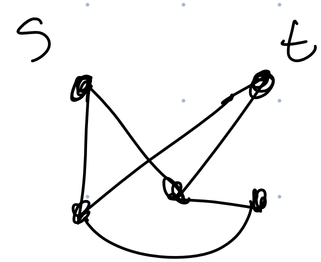
(2) Show $f_{3SAT \to \text{Ham-Path}}$ runs in Polytime

(3.) Show x is 3SAT-Yes iff $f_{3SAT\text{-}\text{Ham-Path}}(x)$ is a Ham-Path-Yes

(1)

$3SAT \leq_p$ HAM-Path:

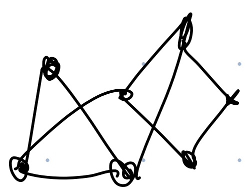
$x = (z_1 \vee \neg z_2 \vee z_3) \wedge (z_3 \vee \cdots) \Longrightarrow$



$\leftarrow x'$ } ?

Know b/c 3-SAT is NP-Hard

{ HAM-PATH $\leq_p$ 3-SAT

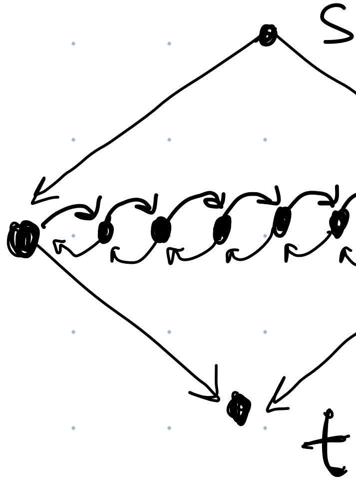


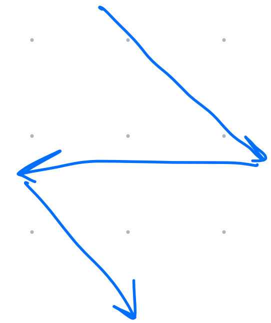
$\Longrightarrow (z_2 \vee z_6 \vee \neg z_8) \wedge \cdots$

How many Hamiltonian Paths are in this graph?

LRL

True



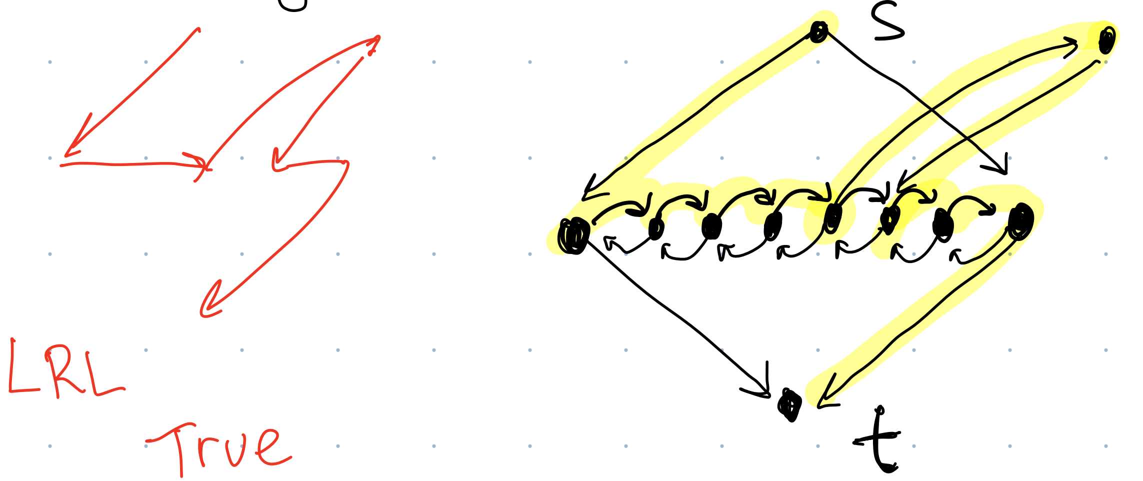


RLR

False

$\boxed{\text{A. } 2}$ B. 3 C. 49 D. $\binom{7}{2}$

How many Hamiltonian Paths are in this graph?

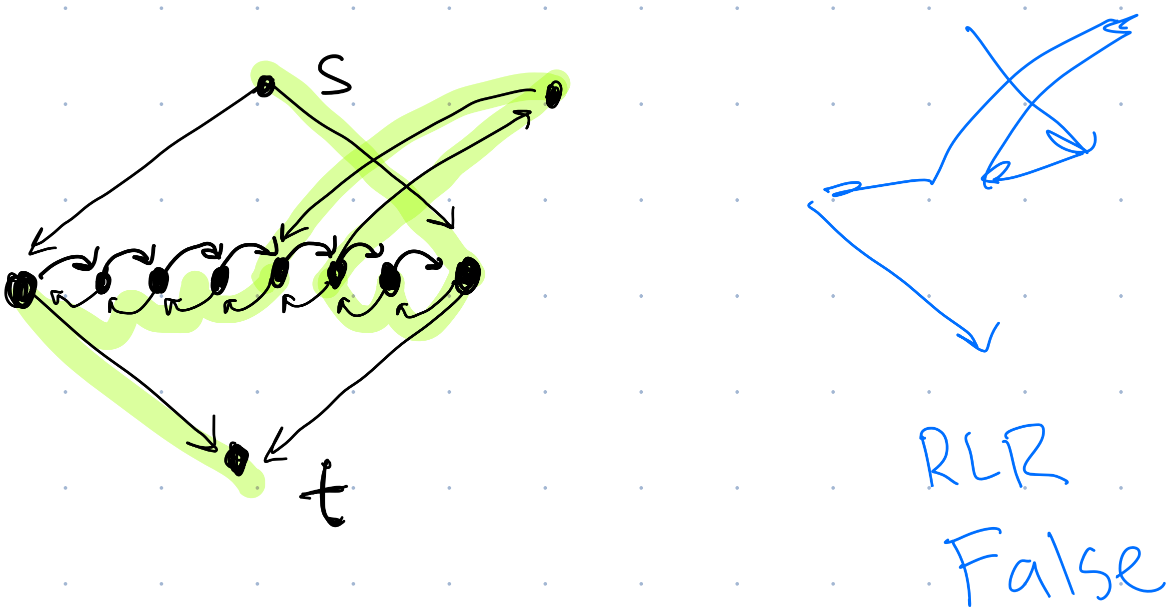


A. 0

B. 1

C. 2

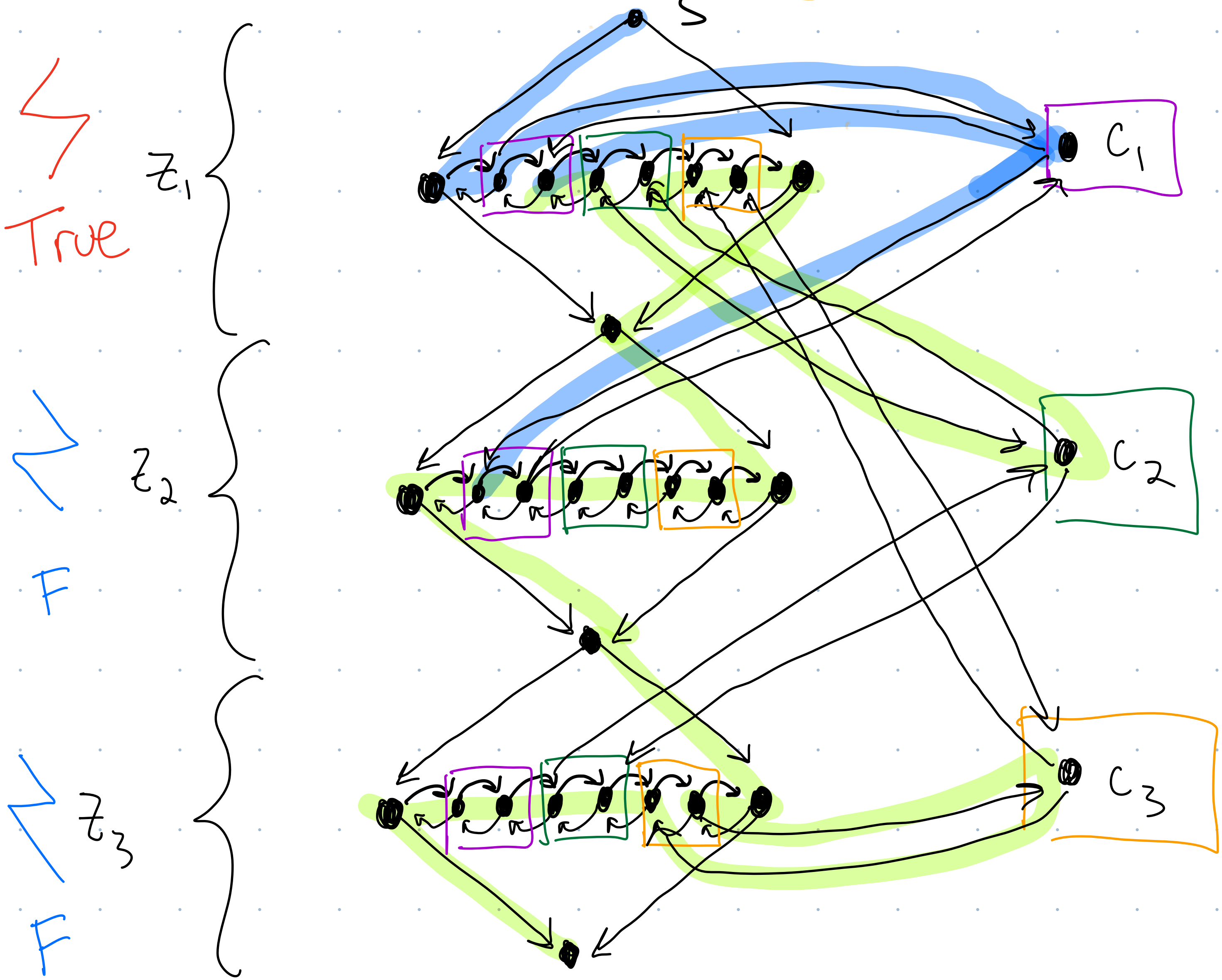
D. 3



$X = \boxed{(z_1 \vee \neg z_2)} \wedge \boxed{(z_1 \vee z_3)} \wedge \boxed{(\neg z_1 \vee \neg z_3)}$

$z_1 = T$

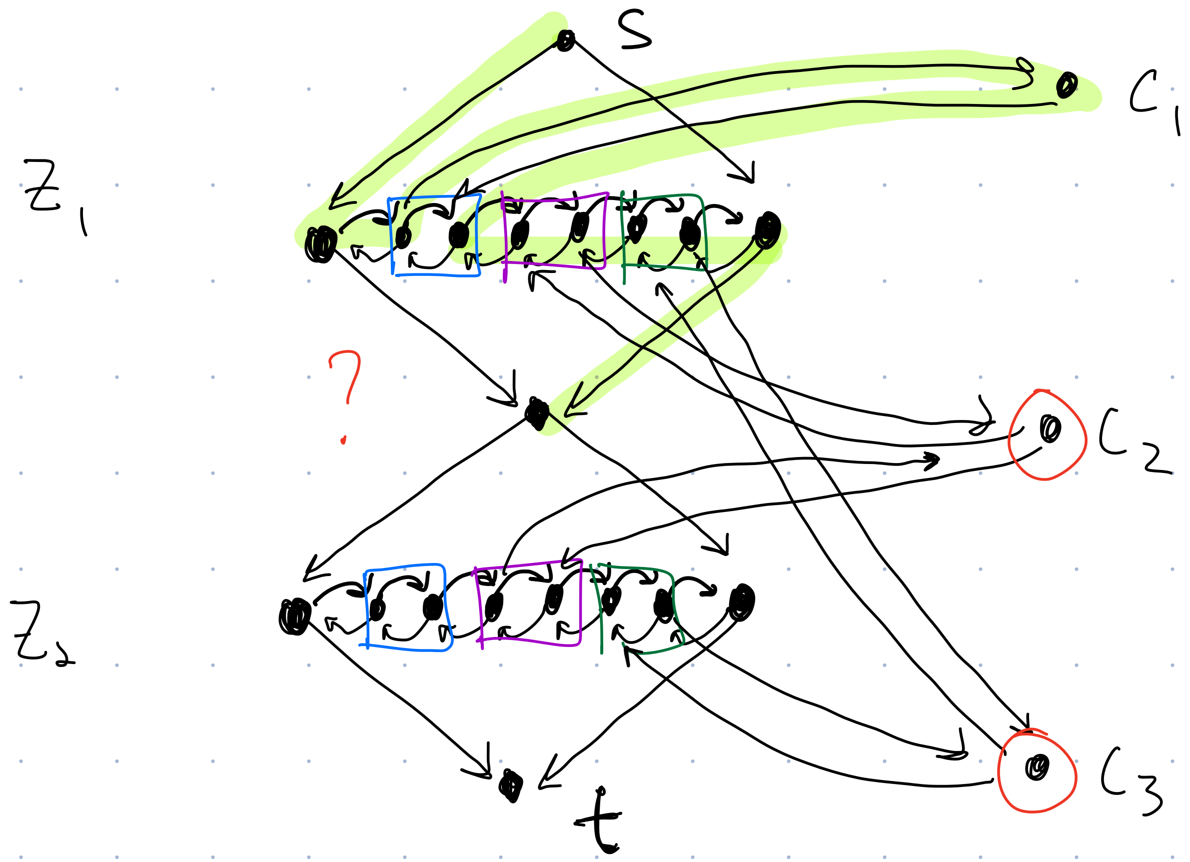
$z_2 = z_3 = F$



Group Work

1. Encode $(z_1) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$ into Ham-Path instance. Show get a No Instance.
2. Runtime of $f_{3SAT \rightarrow HAM\text{-}PATH}$? (Create adj matrix for graph)
3. $3SAT(x) = Yes$ iff $HAMPATH(f_{3SAT-HAMPATH}(x)) = Yes$

1. $(z_1) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$



No Path!

2. Let m = # clauses

n = # variables

$$m \le |X| \lesssim 3m$$

$$n \le |X| \le 2^3 \binom{n}{3} = O(n^3)$$

$$|X| = \text{polynomial}(m, n)$$

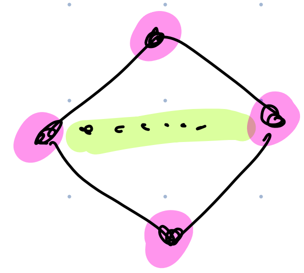
Each gadget: $2m + 4$ vertices

Total gadgets: n

Clause vertices: m

$$\begin{array}{r} n(2m+4) \\ + m \\ \hline 2nm + 4n + m \end{array}$$

$2nm + 4n + m$ vertices $O(mn)$



Adj Matrix: $O(n^2 m^2)$ size

	s	v	u	t
s	0	1		
v				
u				
t				

Filling out array take $O(n^2 m^2)$ time, which is polynomial in n and m.

3.

$$3SAT(x) = Yes \text{ iff } HAMPATH(f_{3SAT-HAMPATH}(x)) = Yes$$

$$(z_1 \lor \overset{T}{\overset{\uparrow}{z_2}} \lor \neg \overset{F}{\overset{\uparrow}{z_3}})$$

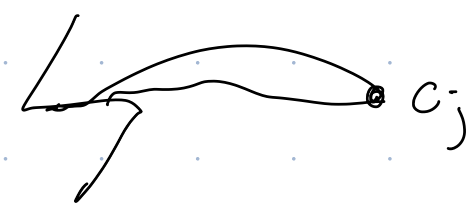
$\Rightarrow$ If $3SAT(x) = Yes$, then there is a satisfying assignment $z_1 \to T$, $z_2 \to F$, ...

Choose one satisfying literal for each clause. Go LRL or RLR through each gadget according to the satisfying assignment, and if that variable is the chosen one for satisfying a clause, jump from gadget to correspending clause vertex, without breaking LRL/RLR flow. In this way we will touch each vertex once. Thus $HAMPATH(f_{SAT \to HAMPATH}(x)) = YES$.

$\Leftarrow$ If $f_{3SAT \to HAMPATH}(x)$ is YES for HAM-PATH, the path must go LRL or RLR through each gadget and when the path jumps to a clause vertex, it must return to same gadget, or otherwise the path would miss vertices.

Then assign $z_i = T$ if LRL thru gadget i
$z_i = F$ if RLR " "

Then this will will satisfy all clauses because each clause will be satisfied by the variable associated with the gadget from which the clause is visited



Note:

Lemma 1: If $Q \in$ NP-Hard and $Q \leq_p R$ then $R \in$ NP-Hard.

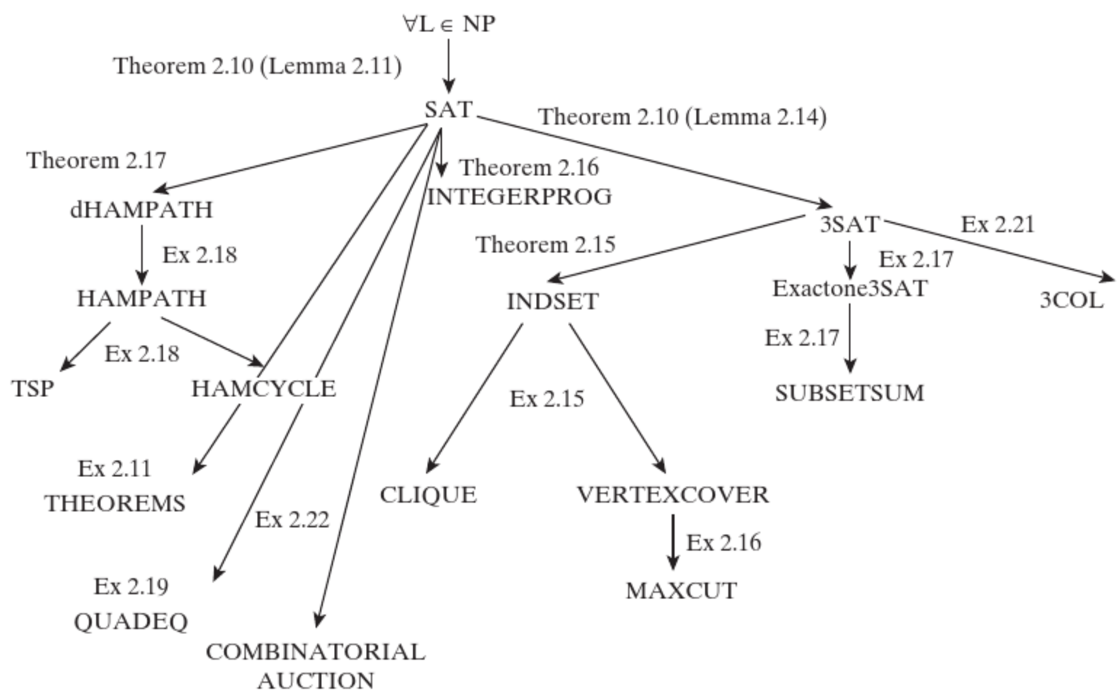


Figure 2.4. Web of reductions between the **NP**-completeness problems described in this chapter and the exercises. Thousands more are known.

(Arora + Boaz, Computational Complexity)