Learning Goals

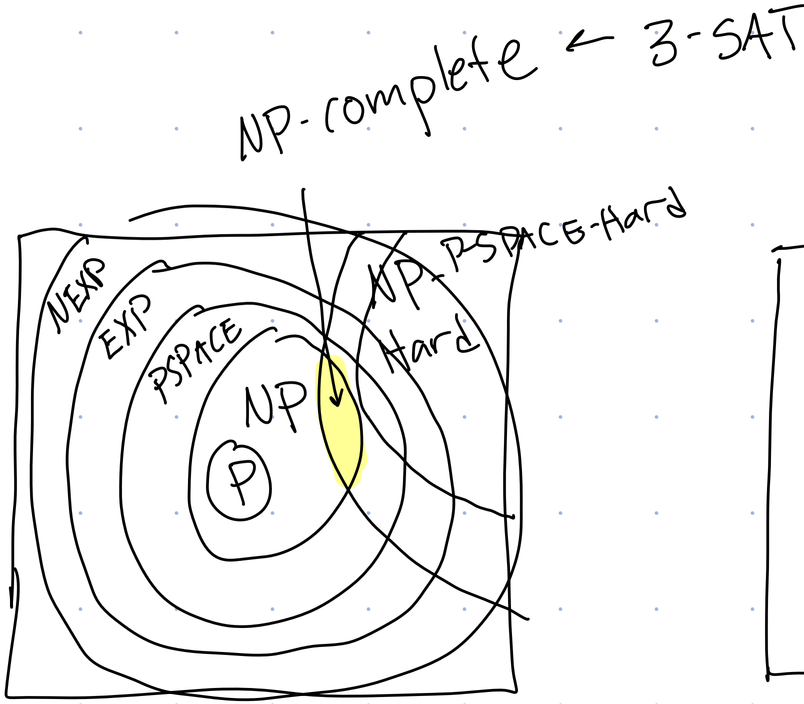
- ✓ Define NP-complete and NP-Hard Problems and describe their importance
- ✓ Describe parts of NP-Complete Proof
- Practice proving a problem is NP complete (Hamiltonian Path)

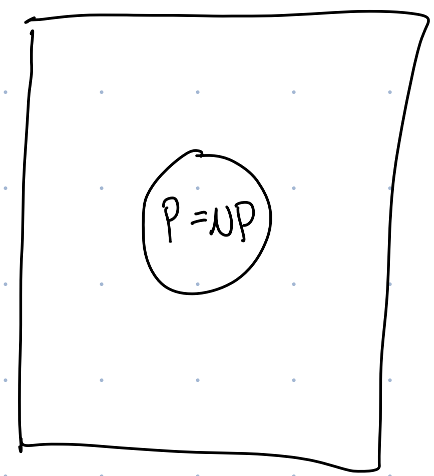
Exit Tickets

- Distributed Bellman-Ford
- Def: NP-Hard? vs. Def NP? vs. Def NP-Complete
- NP-Hard hierarchy
- "Better" polynomial time reductions
- SAT competitions
- Direction of reduction
- $NP \subseteq P$?

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.

Quantum

Reg.

Puzzles / NP

Crossword

Sudoku

Delivery rt ≤ 100 miles

Protein Folding

Factor larger numbers

Primality Testing

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

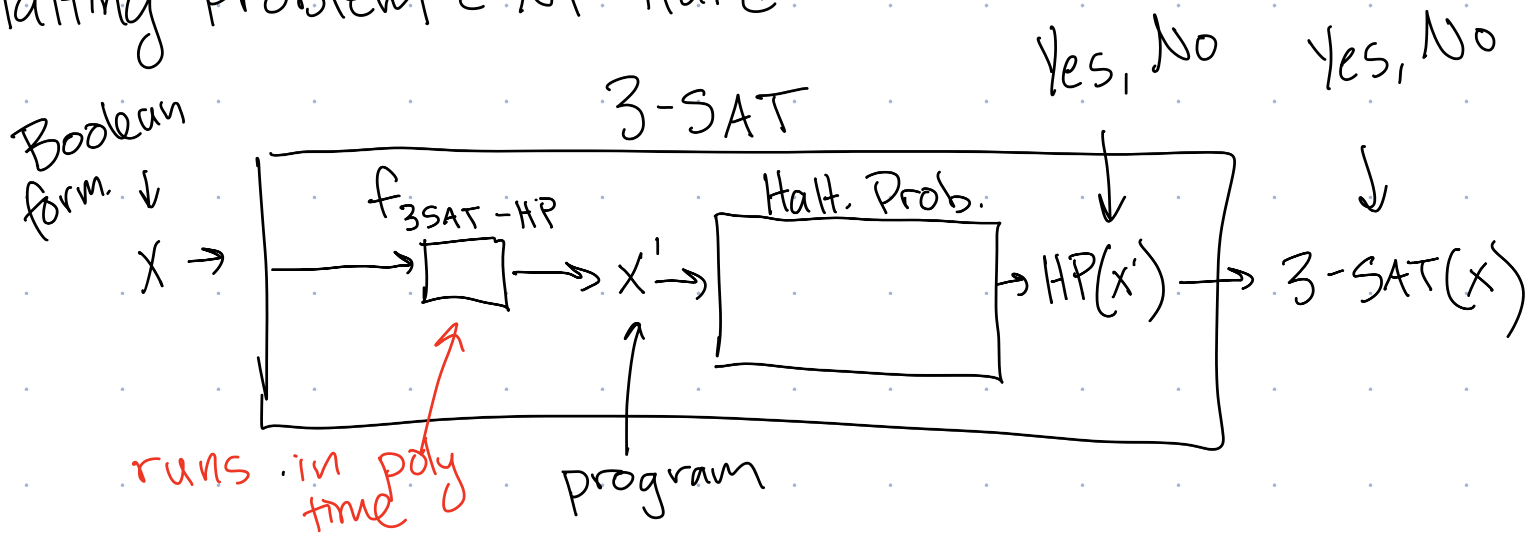
→ Empirical: Keep trying to find alg... but can't... HARD

→ Analytical: Prove a problem is hard. Possible!

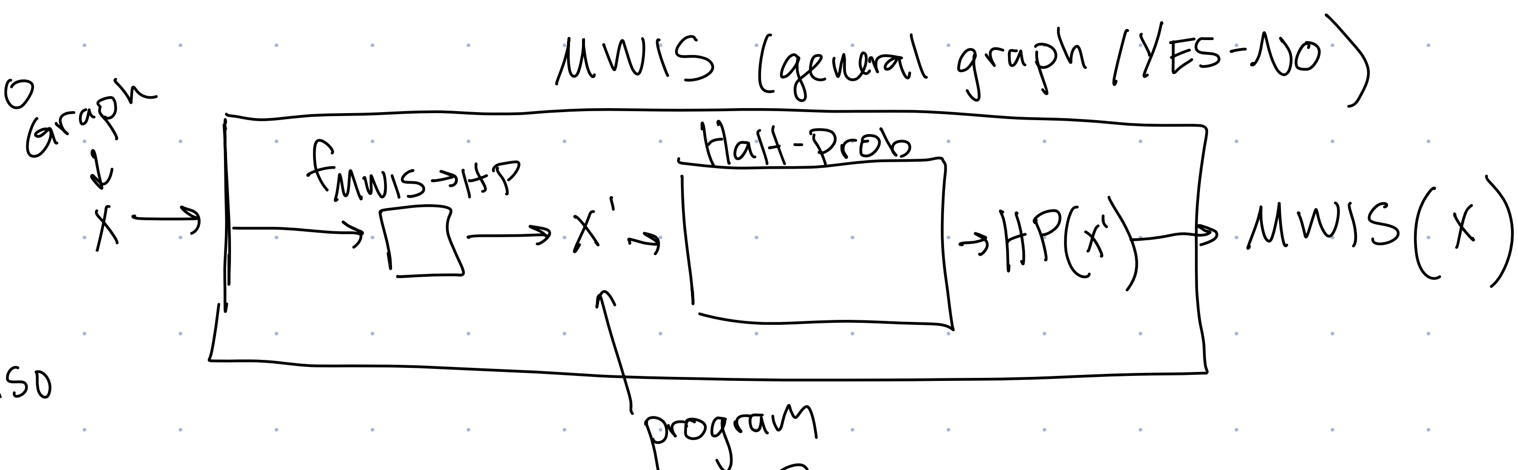
NP-Hard

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

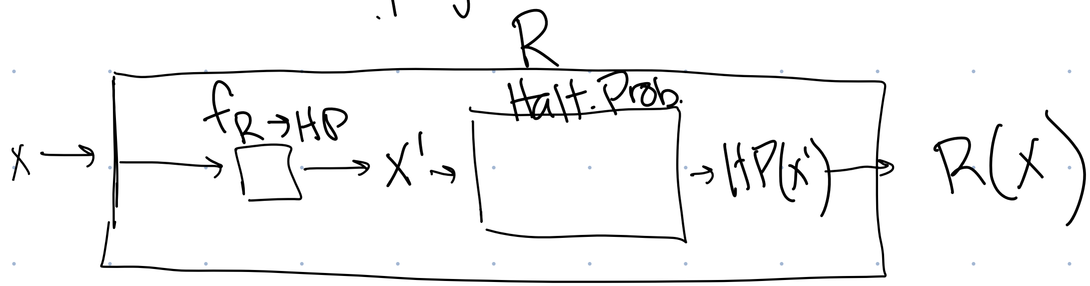
Ex: Halting Problem $\in$ NP-Hard



Also

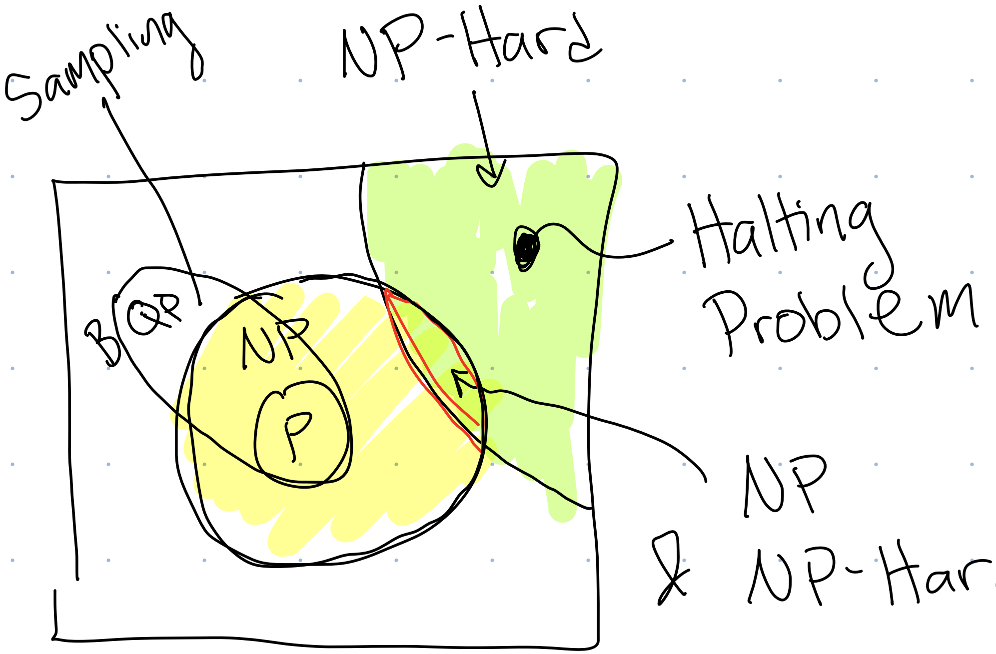


Also



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.

BUT



NP & NP-Hard } def: NP-Complete

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

def: $Q \in$ NP-Complete if $Q \in NP$ and $Q \in$ NP-Hard

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

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

(pset 10)

Theorem: Hamiltonian Path is NP-Complete

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

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

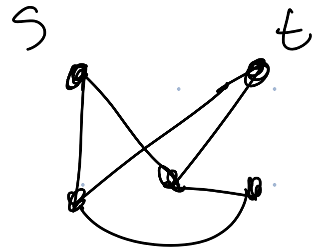
② Show $f_{3SAT \to \text{Ham-path}}$ runs in Polytime

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

①

3SAT $\le_p$ HAM-Path:

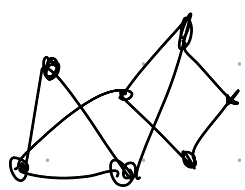
$x = (z_1 \vee \neg z_2 \vee z_3) \wedge (z_3 \vee \ldots) \Longrightarrow$



$\leftarrow x'$?

Know b/c 3-SAT is NP-Hard

HAM-PATH $\le_p$ 3-SAT

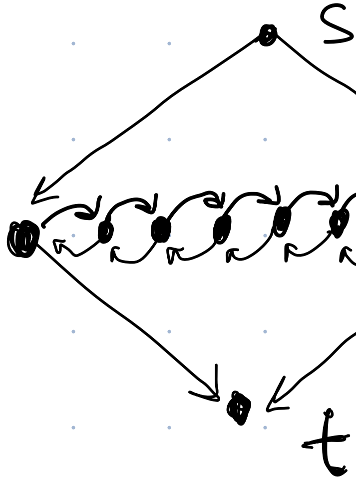


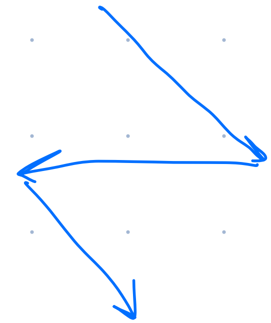
$\Longrightarrow (z_2 \vee z_6 \vee \neg z_8) \wedge \ldots$

How many Hamiltonian Paths are in this graph?

LRL

True





RLR

False

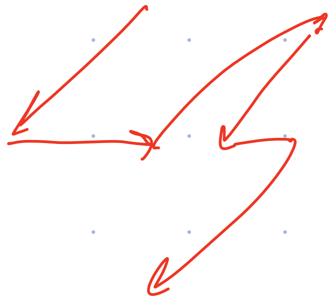
$\boxed{\text{A. } 2}$

B. 3

C. 49

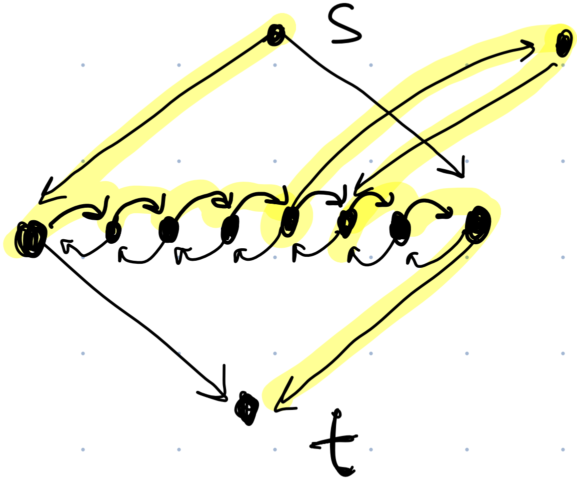
D. $\binom{7}{2}$

How many Hamiltonian Paths are in this graph?



LRL

True

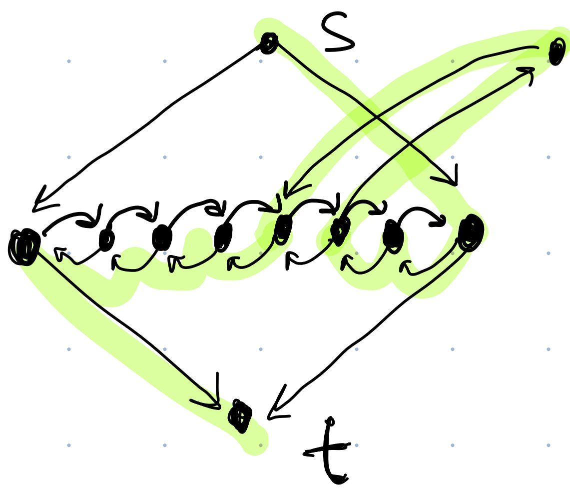


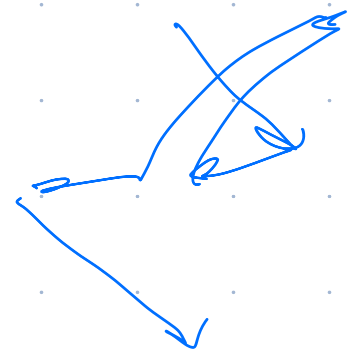
A. 0

$\boxed{\text{B. } 1}$

C. 2

D. 3





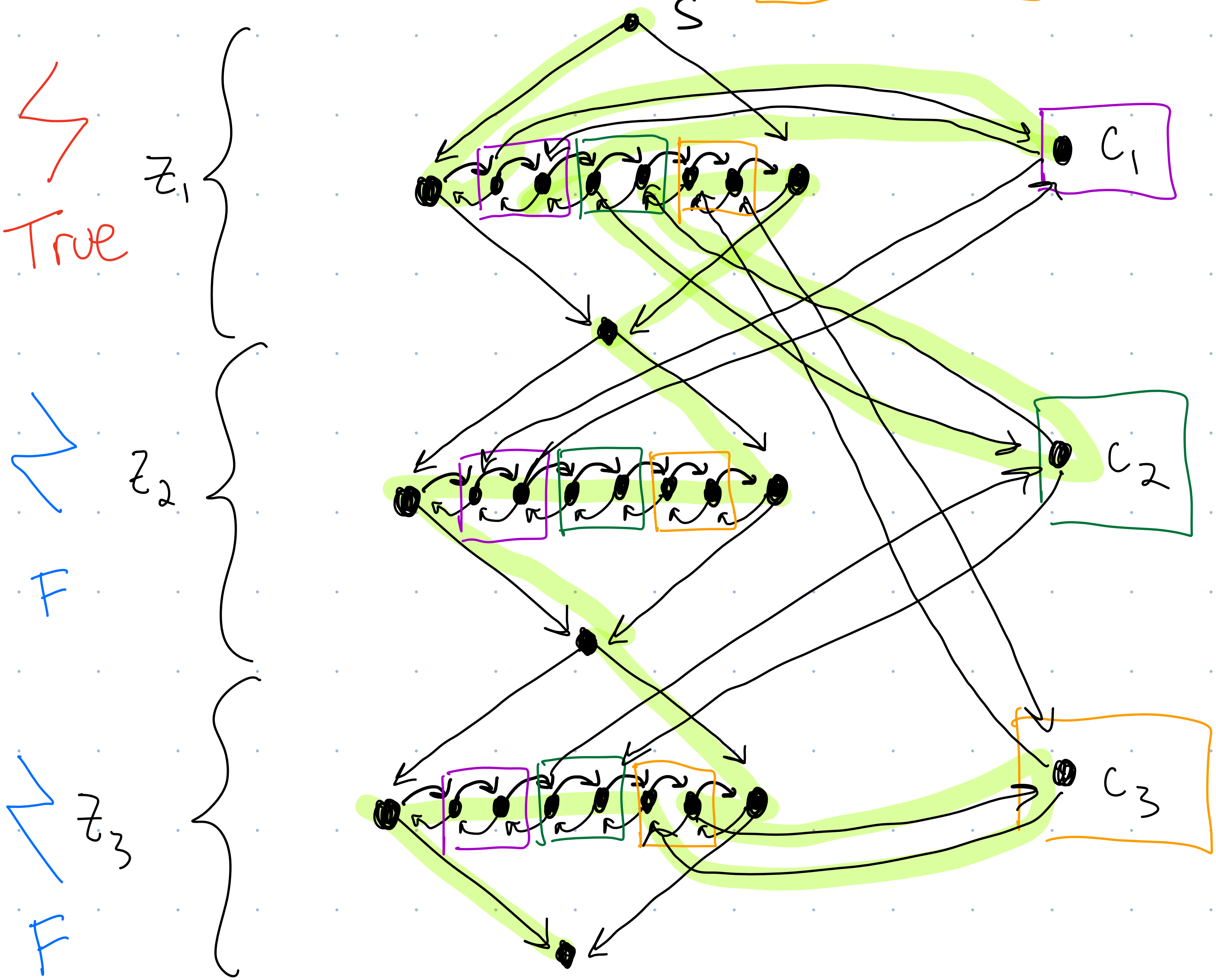
RLR

False

$X = \boxed{(z_1 \lor \neg z_2)} \land \boxed{(z_1 \lor z_3)} \land \boxed{(\neg z_1 \lor \neg z_3)}$

$z_1 = T$

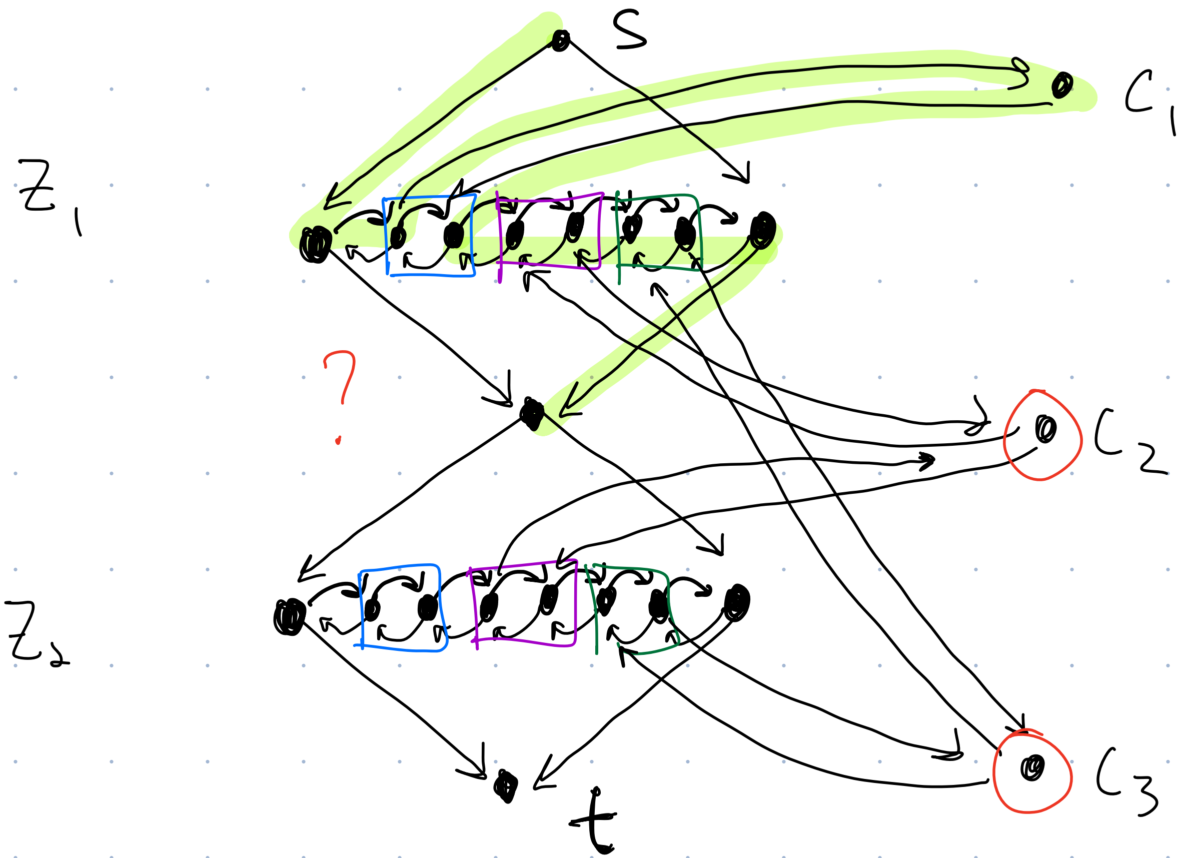
$z_2 = z_3 = F$



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 $f_{3SAT \to HAM\text{-}PATH}$? (Create adj matrix for graph)
3. $3SAT(x) = Yes$ iff $HAMPATH(f_{3SAT-HAMPATH}(x)) = Yes$

1. $(z_1) \land (\neg z_1 \lor z_2) \land (\neg z_1 \lor \neg z_2)$



No Path!

2. Let m = # clauses

n = # variables

size of input to 3-SAT

$m \le |x| \le 3m$

$n \le |x| \le 8\binom{n}{3} = O(n^3)$

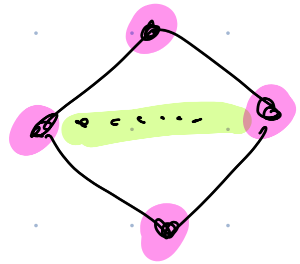
Each gadget: $2m + 4$ vertices

Total gadgets: n

Clause vertices: m

$$n(2m+4) + m$$

$$2nm + 4n + m \text{ vertices}$$



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

~~$O(2^m)$ size~~

$|x| = O(m+n)$

Adj matrix is polynomial in the size of $|x|$, so using for loops, we can fill in this array in poly time

3.

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

$\Rightarrow$ 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 $HAMPATH(f_{SAT \to HAMPATH}(x)) = YES$.

Note:

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

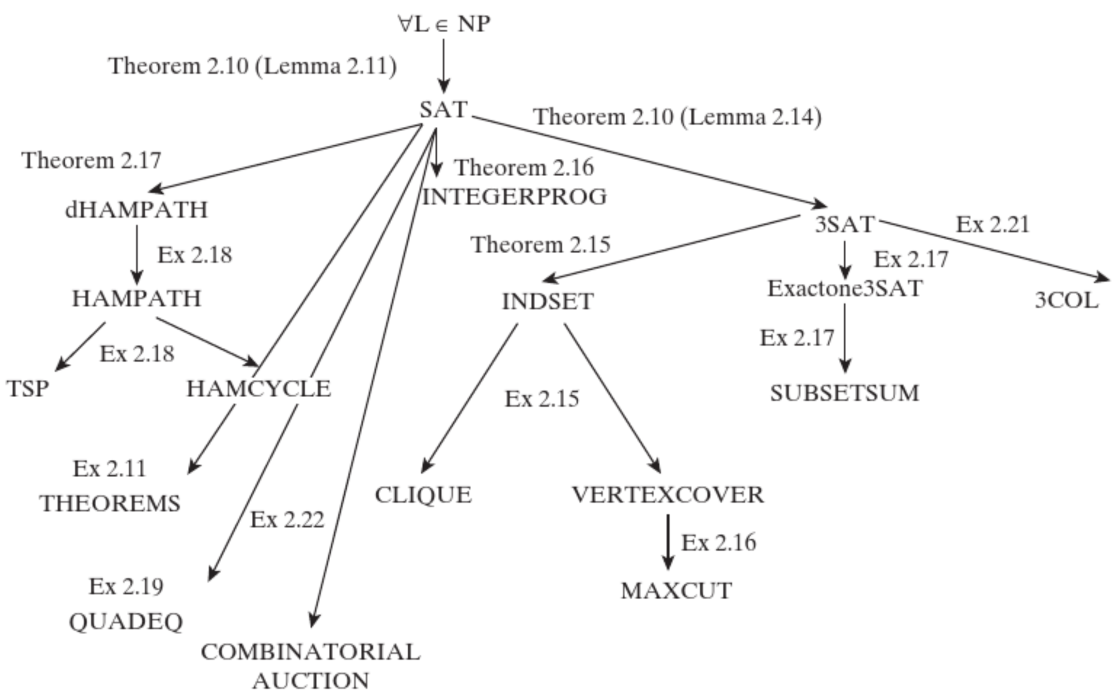


Figure 2.4. Web of reductions between the **NP**-completeness problems described in this chapter and the exercises. Thousands more are known.

(Arora + Boaz, Computational Complexity)