Learning Goals

- Define NP-complete and NP-Hard Problems and describe their importance
- Prove a problem is NP-Hard [NP2]

Exit Tickets

- NP2 on exam
- Graph reduction here useful elsewhere?

Review!

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

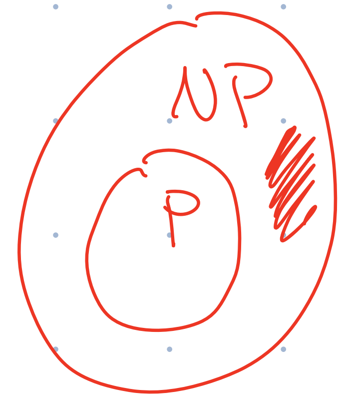
Protein Folding

Factor larger numbers

Primality Testing

Quantum

Reg.



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

→ Empirical: If keep trying to find an alg, but can't...

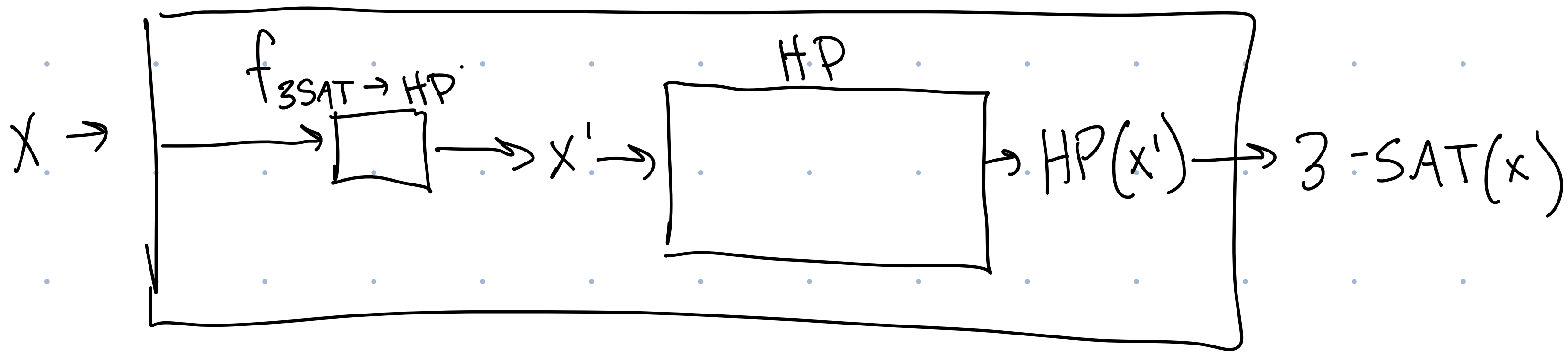
→ Analytical: Can we prove a problem is hard? ☆

NP-Hard

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

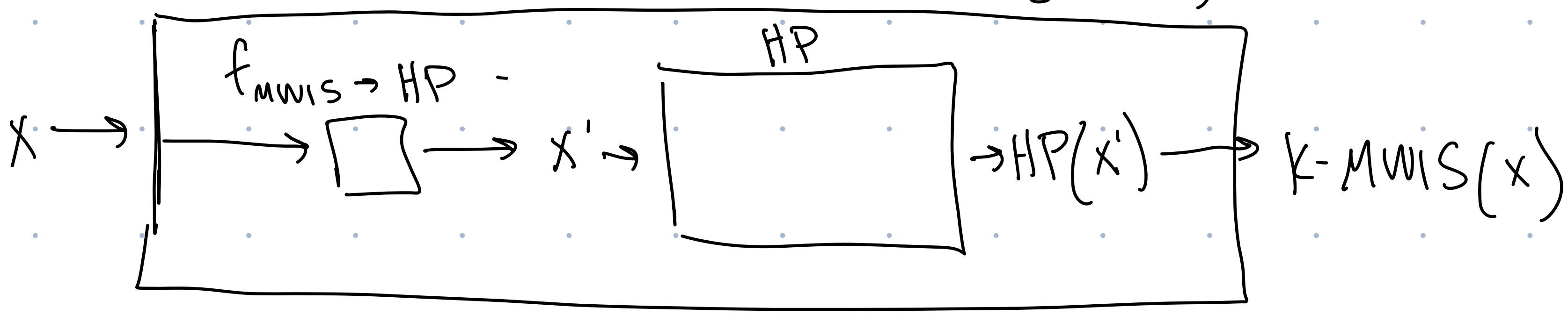
Ex: Halting Problem (HP) $\in$ NP-Hard so

3-SAT



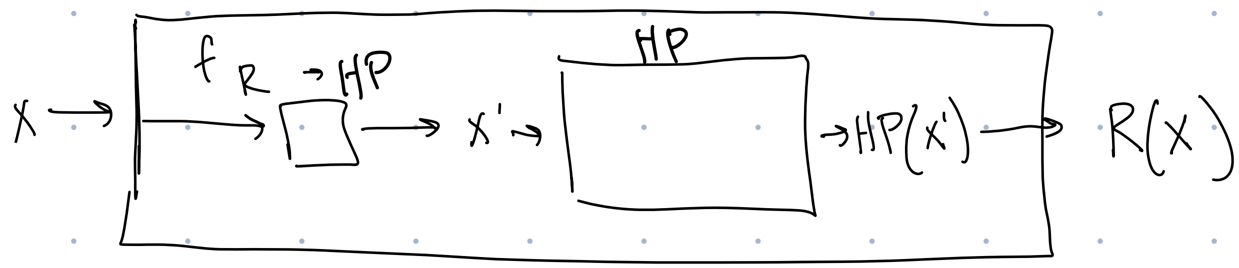
Also

K-MWIS (general graph)



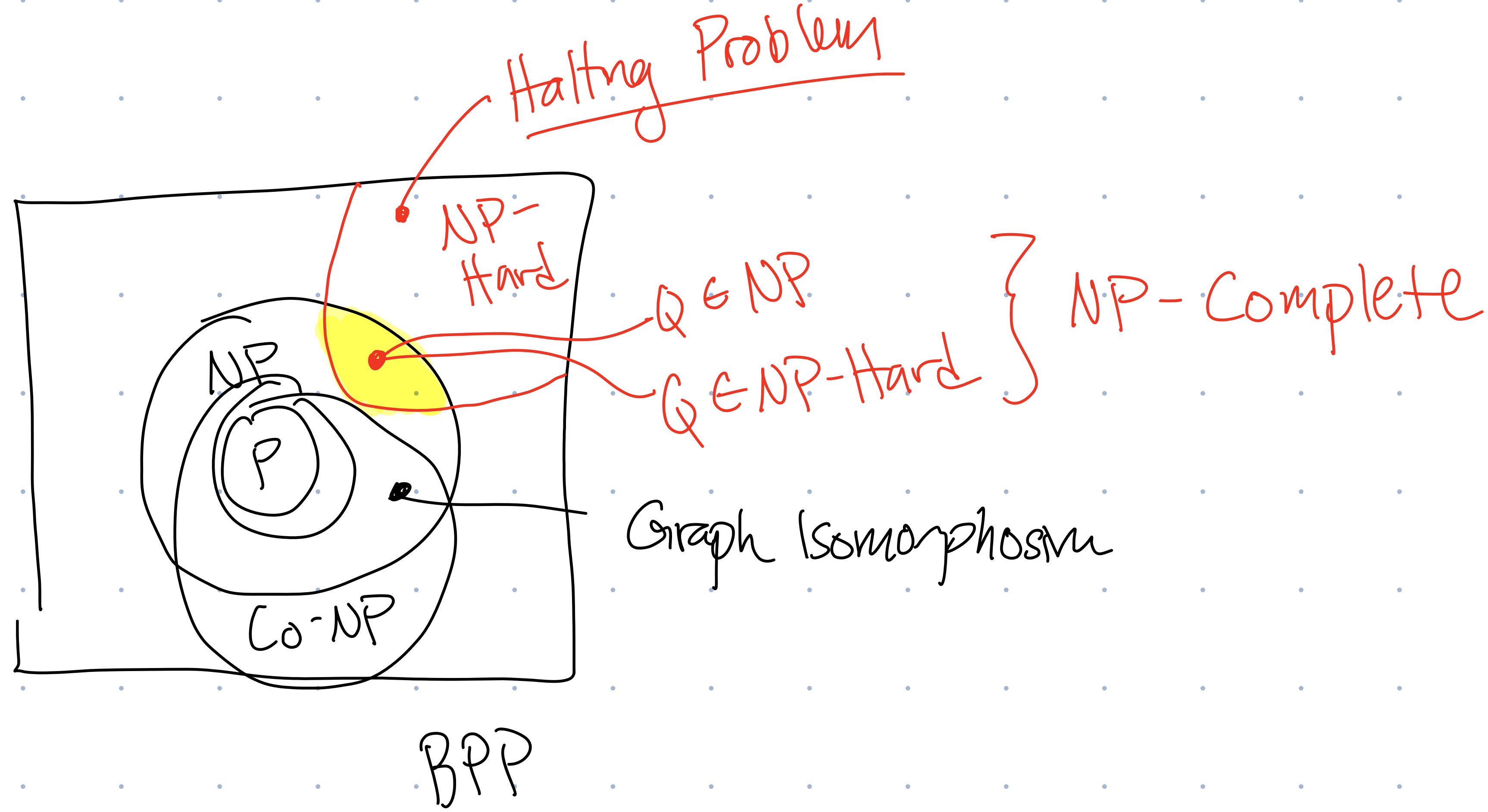
Also For any $R \in NP$, $\exists$ $f_{R \to HamPath}$

R



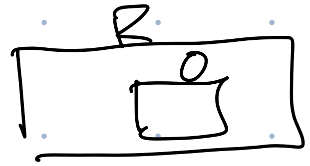
→ NP-Hard problems are harder / require more resources than NP problems, b/c give power to solve all in NP

<u>BUT</u>



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

"R reduces to Q"



Fact 1: $3SAT \in$ NP-Hard (see 301)

3-SAT ≤p HamPath

HamPath

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

3-SAT

NP-Hard Bootstrapping Lemma (See pset 9)

Theorem: Ham-Path is NP-Complete

Pf
- Ham-Path $\in$ NP [insert proof here... see NP class]
- Ham-Path $\in$ NP-Hard

Will Prove: $3SAT \leq_p$ HamPath

+

Fact 1

+

NP-Hard Bootstrapping Lemma

Formal Definition of Polytime Reduction

$\{0,1\}^*$ = set of all bitstrings

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

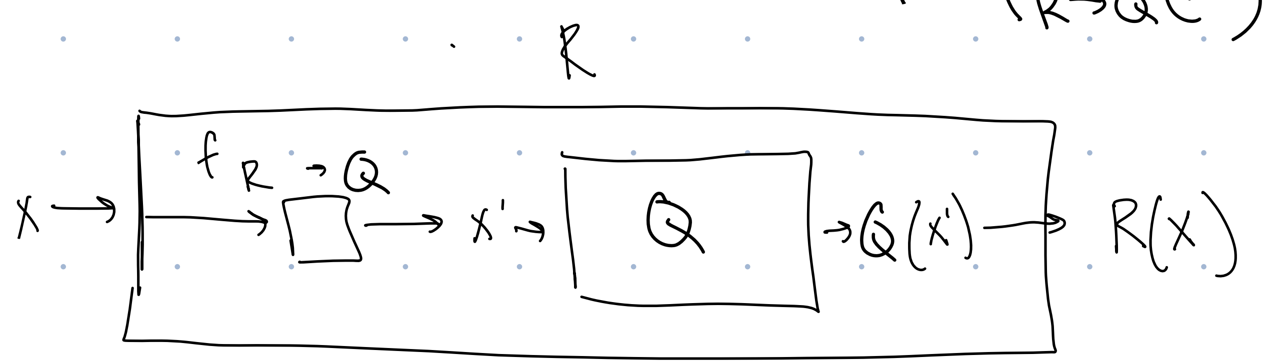
$\exists\, f_{R\to Q} : \{0,1\}^* \to \{0,1\}^*$, s.t. 1

- $\forall x \in \{0,1\}^*$, $R(x) = \text{Yes}$ iff $Q(f_{R\to Q}(x)) = \text{Yes}$ 2

- $\exists$ constant $c_{R\to Q}$ s.t. runtime of $f_{R\to Q}$ on input x is $O(|x|^{c_{R\to Q}})$ 3

$f_{R\to Q}$ runs in polynomial time

$x' = f_{R\to Q}(x)$

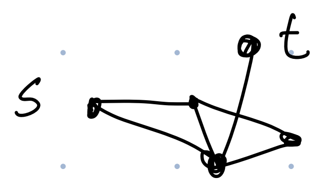


Lemma: $3SAT \le_p \text{Ham-Path}$

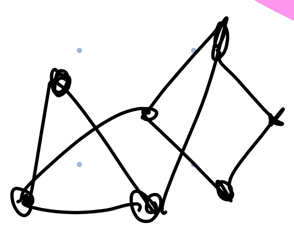
Strategy: 1. Describe $f_{3SAT \to \text{Ham-Path}}$ (turn $3SAT \to \text{Ham-Path}$)

2. Show X is 3SAT-Yes iff $f_{3SAT \to HP}(X)$ is Ham Path-Yes

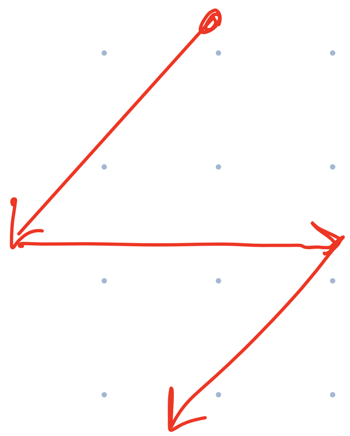
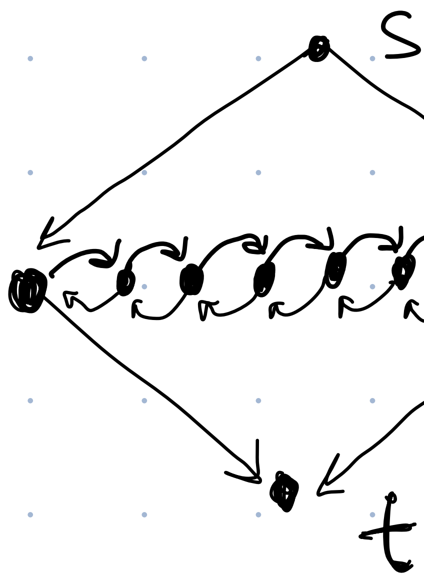
3. Show $f_{3SAT - \text{Ham-Path}}$ polytime

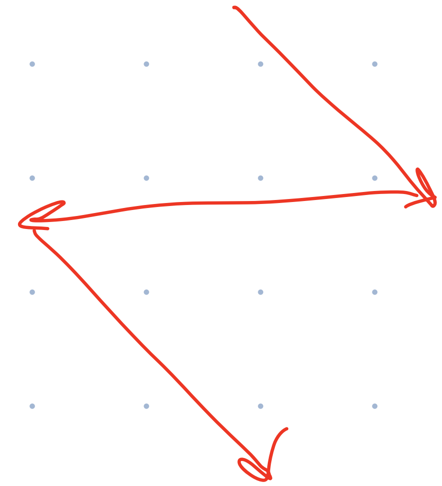
1. $(z_1 \vee \neg z_2 \vee z_3) \wedge \cdots \implies$ 

Common mistake!

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

How many Hamiltonian Paths are in this graph?



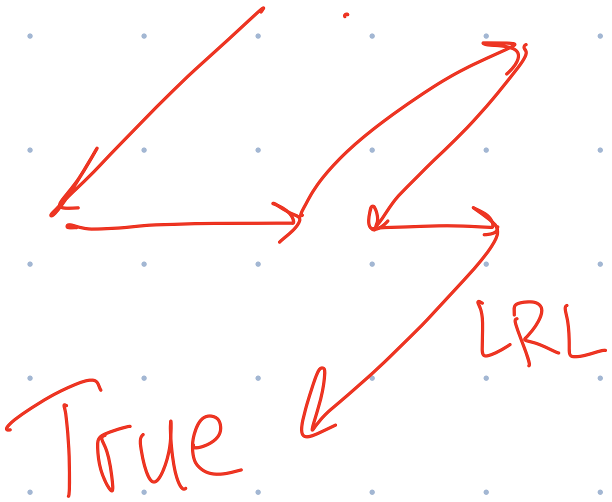
A. 2

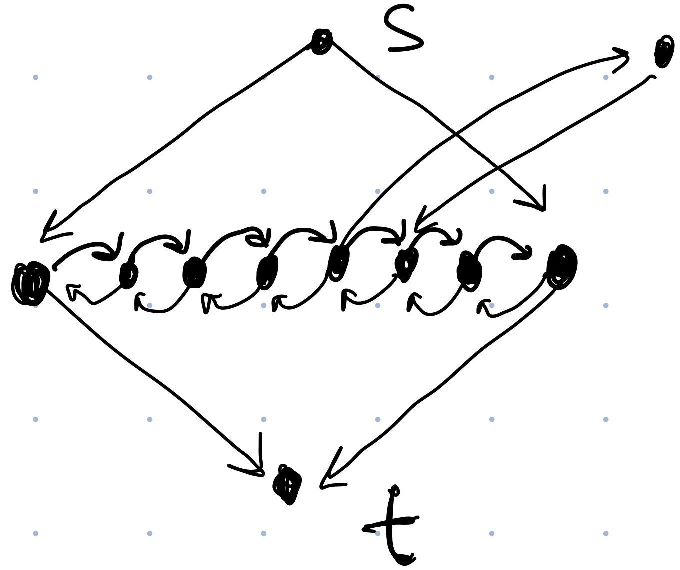
B. 3

C. 49

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

How many Hamiltonian Paths are in this graph?



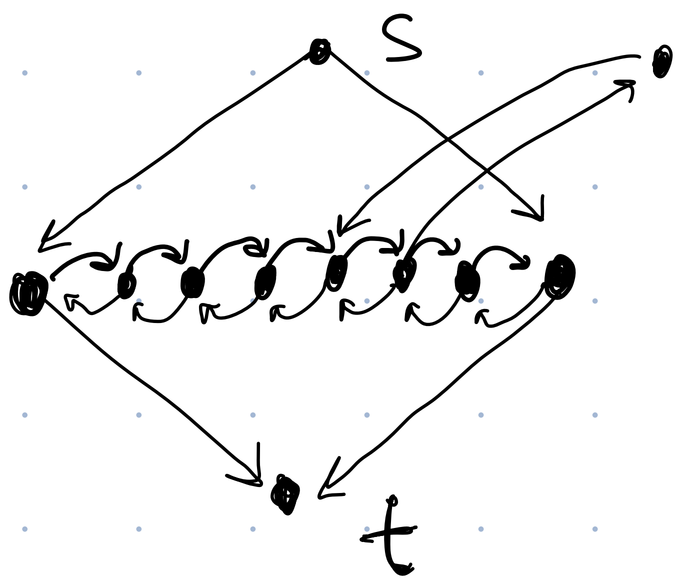


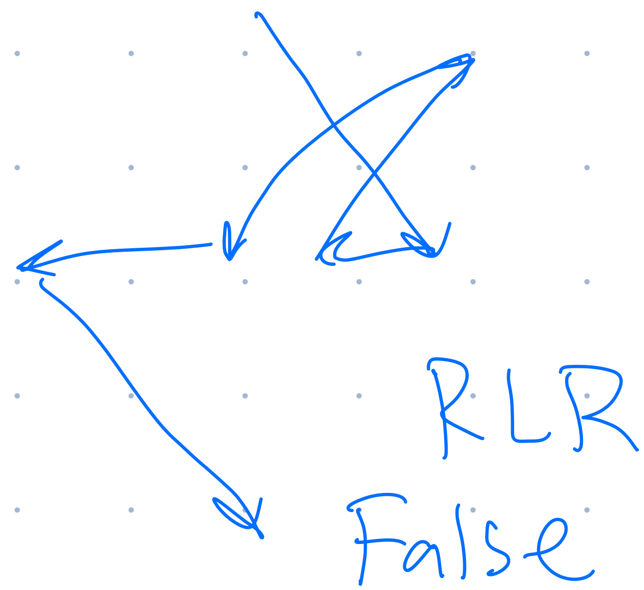
A. 0

B. 1

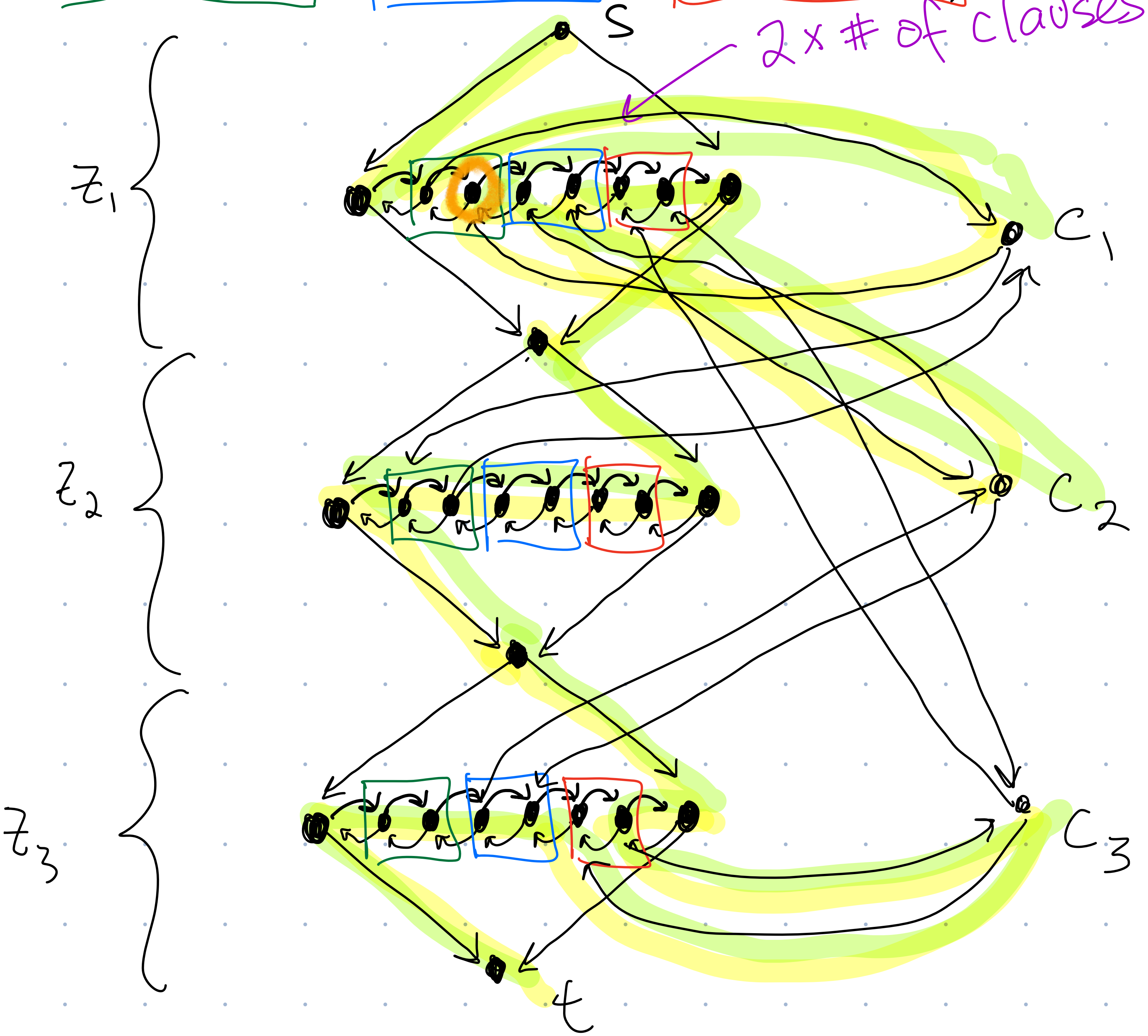
C. 2

D. 3





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



TRUE

False

$z_1 = T$

$z_2 = F$

$z_3 = F$

Group Work

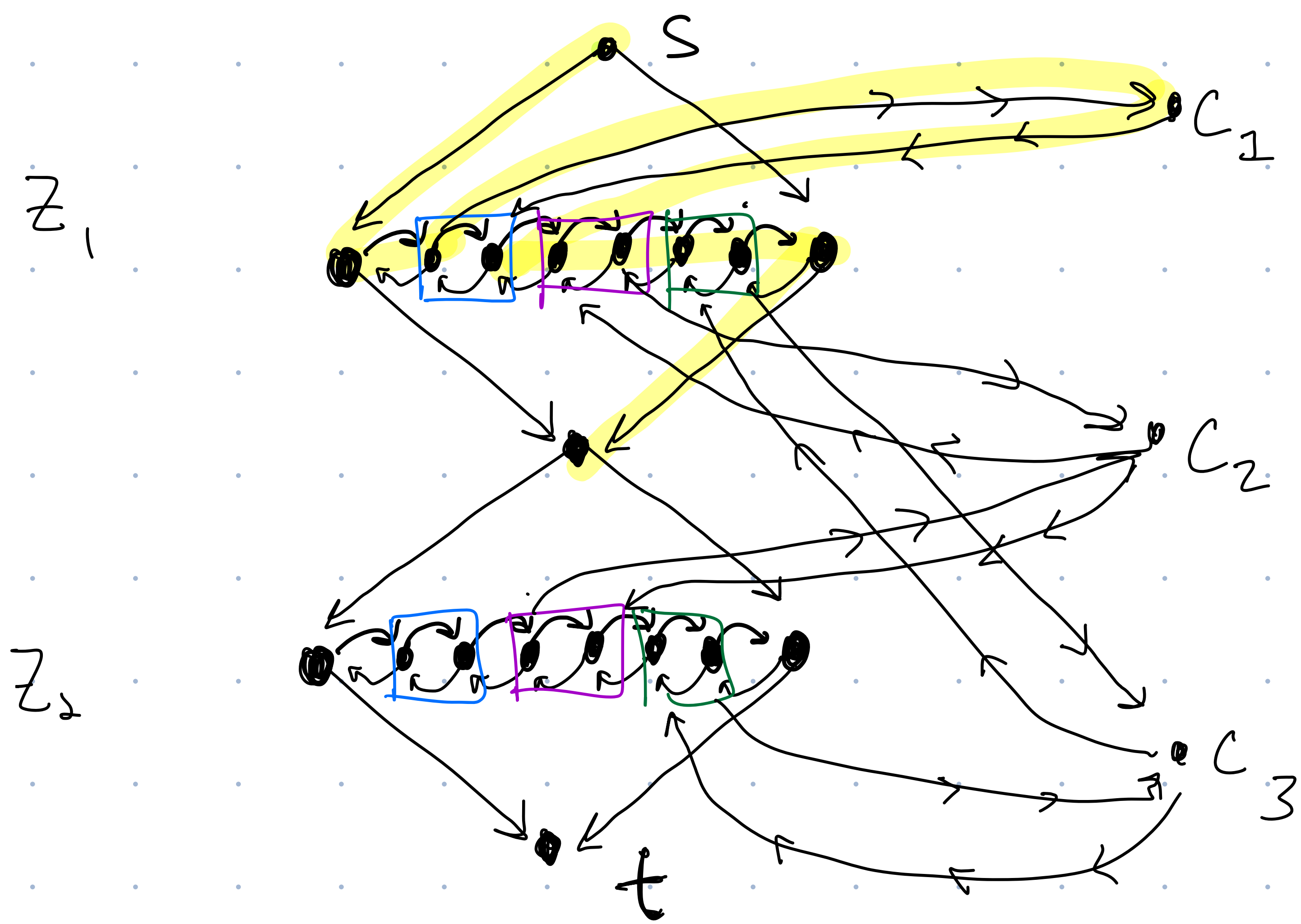
1. Encode $(z_1) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$ into Ham-Path instance. Show get a No Instance.
2. Runtime of $f_{3SAT \to HAM\text{-}PATH}$? (Create adj matrix for graph)
3. $3SAT(x) = \text{Yes}$ iff $HAMPATH(f_{3SAT-HAMPATH}(x)) = \text{Yes}$

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

$z_1 = T$

$z_2 = T \quad \neg z_2 \, T \quad X$

No sat. assignment



X No path

2.

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

$\Rightarrow$ If $3SAT(x) = \text{Yes}$, then there is a satisfying assignment

$z_1 \to T$
$z_2 \to F$
$\vdots$

where each clause has at least one satisfying literal. Choose one satisfying variable for each clause. Go LRL or RLR through each gadget according to the truth value of the chosen variable. Then it is possible to jump out to the clause vertex without breaking LRL, RLR flow. In this way we will touch every vertex once.

$\Leftarrow$ If $f_{3SAT \Rightarrow HAMPATH}(x)$ is yes for Ham Path, any path must go LRL or RLR through each gadget. When the path goes out to a clause vertex, it must return to the same gadget or otherwise we would repeat vertices (see video for more explanation). If we assign $z_i = T$ if LRL in gadget i
$z_i = F$ if RLR in gadget i
Will satisfy all clauses since each clause will be satisfied by variable associated with gadget from which the clause vertex is visited.

3. Let $m = \#$ clauses

$n = \#$ variables

vertices

Each gadget: $2m+4$ vertices

Total gadgets: n

Clause vertices: m

$$\left.\begin{array}{l}\text{Each gadget: } 2m+4 \text{ vertices}\\ \text{Total gadgets: } n\\ \text{Clause vertices: } m\end{array}\right\} n(2m+4)+m = O(nm)$$

Adjacency matrix: $O(n^2m^2)$

Total edges: Gadget: $O(m)$

Clauses: $O(m)$

$|x| = \Omega(m)$

Polynomial time in $|x|$

Note: Once we prove Ham-Path ∈ NP-Hard, we can combine with Lemma 1 to prove new problems are NP-Hard

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

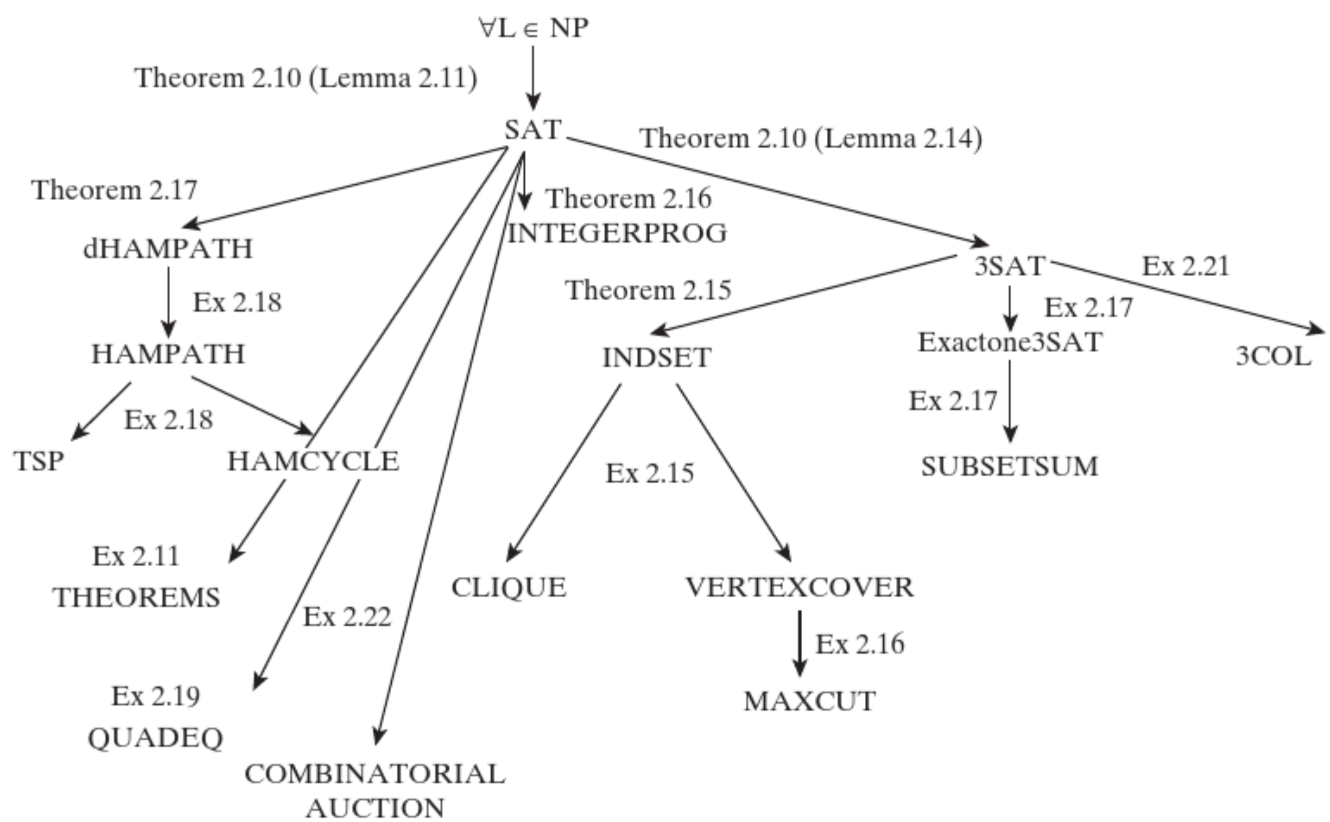


Figure 2.4. Web of reductions between the **NP**-completeness problems described in this chapter and the exercises. Thousands more are known.

(Arora + Boaz, Computational Complexity)